APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



JPRS L/10387

15 March 1982

# USSR Report

## CYBERNETICS, COMPUTERS AND AUTOMATION TECHNOLOGY

(FOUO 3/82)

FBIS FOREIGN BROADCAST INFORMATION SERVICE



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

NOTE

JPRS publications contain information primarily from foreign newspapers, periodicals and books, but also from news agency transmissions and broadcasts. Materials from foreign-language sources are translated; those from English-language sources are transcribed or reprinted, with the original phrasing and other characteristics retained.

Headlines, editorial reports, and material enclosed in brackets [] are supplied by JPRS. Processing indicators such as [Text] or [Excerpt] in the first line of each item, or following the last line of a brief, indicate how the original information was processed. Where no processing indicator is given, the information was summarized or extracted.

Unfamiliar names rendered phonetically or transliterated are enclosed in parentheses. Words or names preceded by a question mark and enclosed in parentheses were not clear in the original but have been supplied as appropriate in context. Other unattributed parenthetical notes within the body of an item originate with the source. Times within items are as given by source.

The contents of this publication in no way represent the policies, views or attitudes of the U.S. Government.

COPYRIGHT LAWS AND REGULATIONS GOVERNING OWNERSHIP OF MATERIALS REPRODUCED HEREIN REQUIRE THAT DISSEMINATION OF THIS PUBLICATION BE RESTRICTED FOR OFFICIAL USE ONLY.

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



JPRS L/10387

15 March 1982

# USSR REPORT

## CYBERNETICS, COMPUTERS AND AUTOMATION TECHNOLOGY

(FOUO 3/82)

## SELECTED ITEMS FROM 'COMPUTER TECHNOLOGY OF THE SOCIALIST COUNTRIES', No. 9, 1981

## CONTENTS







APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6







APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



COMPUTER TECHNOLOGY OF THE SOCIALIST COUNTRIES

Moscow VYCHISLITEL'NAYA TEKHNIKA SOTSIALISTICHESKIKH STRAN in Russian No 9, 1981 (signed to press 21 May 81) pp 2-4, 172-173, 176-179

[Annotation, foreword, table of contents and abstracts from book "Computer Technology of the Socialist Countries", edited by M. Ye. Raykovskiy, a collection of articles, Izdatel'stvo "Finansy i statistika", 16,000 copies, 184 pages]

[Text] This international collection deals with problems of research, development, application and operation of computer facilities developed in accordance with the agreement on cooperation in computer technology between the NRB [People's Republic of Bulgaria], the VNR [Hungarian People's Republic], the GDR [German Democratic Republic], the PNR [Polish People's Republic], the Republic of Cuba, the SRR [Socialist Republic of Romania], the USSR and the ChSSR [Czechoslovak Socialist Republic].

In the collection, special attention is paid to problems of computer applications in ASU [automated management systems] and computer-aided design systems.

The collection is addressed to workers engaged in development and use of the facilities in the Unified System of Computers in various sectors of the national economy.

Foreword

Since 1979, YeS computer users have been receiving computers in the second phase of the Unified System. Industry has now completed fully the transition to production of these new machines. Their functional capabilities are raising the technical and economic effect from computer applications in the most varied spheres of the national economy of the socialist countries. This is facilitated also by the new operating systems that permit implementation of the time-sharing mode in shared-use systems, virtual distribution of machine resources, real-time control of objects, etc. The capabilities of the new machines and the systems based on them have been expanded through an increase in main storage capacity, use of 100M-byte disk storage units, and use of new teleprocessing facilities and specialized processors.

The thematic direction of a number of articles in this collection is the development of the Unified System of Computers. Another part contains articles dealing with computer-aided design [CAD] systems (SAPR) based on using the Unified System and System of Small Computers. CAD systems are one of the most effective



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



applications of computers. CAD permits reducing the time for design of complex objects two to three-fold and raises considerably their technical and economic parameters. Articles on the problems of improving programming are included in this collection.

The remaining articles in the collection pertain to problems of application and operation of the YeS and SM computer hardware and software. Articles on new hardware and software are in the last section.

The editorial board for the collection: M. Ye. Rakovskiy, chief editor (Coordination Center, MPK [Intergovernmental Commission]), N. V. Gorshkov (Council on Complex Maintenance of the YeS EVM), Kh. Choppe (GDR), Ye. N. Mel'nikova, responsible secretary (Coordination Center, MPK), B. N. Naumov (Council of Chief Designers of the SM EVM), L. Nemet (VNR [Hungarian People's Republic]), P. Popov (NRB [People's Republic of Bulgaria]), V. V. Przhiyalkovskiy (Council of Chief Designers of the YeS EVM), Yu. P. Selivanov, editor-in-chief (USSR), A. Ye. Fateyev (USSR) and N. I. Cheshenko (Council on Application of SVT [Computer Technology]).

Contents Page





APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6





Abstracts

TASK OF TECHNICAL AND ECONOMIC JUSTIFICATION IN SELECTING CHARACTERISTICS OF PARAMETRIC SERIES OF COMPUTERS

[Abstract of article by Selivanov, Yu. and Novitskas, Yu.]

[Text] Basic features of a family of computers are considered as a parametric series of machines performing data processing in various applications. Technical and economic criteria for optimization of characteristics of parametric series of computers and ways of solving optimization problems are presented.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



CONCEPT OF YES-1060 COMPUTER DESIGN AND DEVELOPMENT

[Abstract of article by Antonov, V; Orayevskiy, K. and Avtonomov, B.]

[Text] Stage-by-stage development of the YeS-1060 computer, design principles and characteristics of central units are described. Processor structure, providing high efficiency in data processing, and the structure and characteristics of the YeS-4001 universal channel are presented. A properly selected direction for stage-by-stage development of a model, that enables developing a computer with high technical and economic parameters, is shown.

UNIFIED SECONDARY POWER SUPPLIES FOR SERIES-2 UNIFIED SYSTEM COMPUTERS

[Abstract of article by Fayzulayev, B. N.; Batyukov, Ye. I. and Mkrtchyan, Zh. A.]

[Text] Basic specifications for computer secondary power supplies are cited. Modular design of functionally complete power supplies is justified. Basic technical and economic features of the power supplies are given. Design features for power supply systems for hardware in second phase of YeS EVM and problems of using secondary power supplies with input without transformers are considered. Specific characteristics of power supply systems for the first and second phases of the YeS EVM are compared.

DEVELOPMENT OF DATA PREPARATION UNITS IN BULGARIA

[Abstract of article by Topalov, T. A.]

[Text] Basic functional features and technical data of key-to-tape and key-to-disk units are considered. Information of the YeS-9002 and YeS-9003 key-to-tape units and key-to-floppy disk units developed and manufactured in Bulgaria is given.

CHARACTERISTIC FEATURES OF THE YES-1045 COMPUTER

[Abstract of article by Kuchukyan, A. T.; Sarkisyan, T. Ye.; and Mkrtumyan, I. B.]

[Text] Basic parameters of the YeS-1045 computer, set of standard and optional facilities and ways of obtaining new configurations are analyzed. The structure of the CPU and characteristics of its components are given. A highly developed checking and diagnostic system, providing higher reliability and repairability for the machine, is described. System capabilities of the model and organization of the array processor and additional capabilities that result when it is connected are discussed.

VIDEOTON FAMILY OF INTELLIGENT ALPHANUMERIC TERMINALS

[Abstract of article by Nylas, L.; Hudoba, G. and Bornemissza, J.]

[Text] General trends in terminal development are discussed. Methods of developing terminals with alphanumeric displays at the Videoton enterprise are shown. Functional capabilities of new video terminals with microprogram control, based on microprocessors, for the YeS and SM computers are described. Technical characteristics for the SM-7219, YeS-7168N, SM-7401 and UT-20 units are given.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



YES-2335 ARRAY PROCESSOR FOR THE YES-1035 COMPUTER SYSTEM

[Abstract of article by Nikolov, G. P.; Lazarov, V. D.; Daskalov, P. P.; Ivanova, Y. V.; and Kirov, K. D.]

[Text] The YeS-2335 array processor functional capabilities, general structure and methods of connection to the computer are described. It permits a significant increase in computer throughput for special tasks.

SYSTEM FOR DESIGN OF APPLICATION PROGRAM PACKAGES FOR CALENDAR-SUBJECT PLANNING AND MANAGEMENT FOR YES COMPUTERS

[Abstract of article by Matulis, V. A. and Chaplinskas, A. A.]

[Text] Basic system characteristics are given; package design technology is described. Given as an example are program packages for five-year planning of program development in the machine-building sector, and for quarterly and annual planning of program development for enterprises using the system described.

DIGITAL SIMULATION OF CONTINUOUS PROCESS ON YES COMPUTERS

[Abstract of article by Yazhëmbek, Ya.]

[Text] The DIANA problem-oriented programming language, used to describe a system to be simulated in digital form, is presented. The language permits solving problems written in the form of differential equations or an analog block-diagram. The basic functions of the language, all its functional statements and the structure of a DIANA program are given. Advantages of this language compared to other similar ones, CSMP for example, are considered with regard to the need of solving problems on the small YeS models.

PROBLEM-ORIENTED LANGUAGE FOR CAD SYSTEM FOR MANUFACTURING PROCESSES OF MECHANICAL MACHINING OF PARTS

[Abstract of article by Tsvetkov, V. D.; Tolkachev, A. A.; and Cher, I.]

[Text] The language is intended for design of informational models of parts to be processed in systems for CAD of manufacturing processes of mechanical machining of parts. Facilities and rules for formalized description of structural elements and machine parts as well as the dimension and precision links between these elements are given. The language may be used in CAD systems for industrial processes and to prepare control programs for machine tools with numeric control.

EXPERIENCE IN USE OF TIME-SHARING SYSTEM FOR PROGRAM DEVELOPMENT

[Abstract of article by Brusdeylins, G.]

[Text] A year's experience in using facilities of a time-sharing system to improve development of problem-oriented software is described. The individual stages of program compilation in the time-sharing mode are discussed in detail; technical and economic effects are evaluated.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



SZPAK-77 INTEGRATED PROGRAMMING SYSTEM FOR COMPLEX AUTOMATION

[Abstract of article by Aderek, A.]

[Text] The basic elements of the SZPAK-77 system are presented. It is a form system of programming. Its language allows convenient writing of data acquisition and preprocessing algorithms. Process variable processing algorithms can be extended by using auxiliary programs in the extended RT-FORTRAN-IV-E language. The SZPAK-77 system has an interactive language for communication with the operator to input changes into the system, expand it, initiate standard reports, etc.

DOS RT REAL-TIME OPERATING SYSTEM

[Abstract of article by Shuba, A.]

[Text] The features, purpose and degree of complexity of the DOS RT for the SM-3 and SM-4 minicomputers are described. The system offers the capability of creating an environment of several concurrently operating program processes that interact with each other and with interactive operators and external processors.

SYSTEMATIC DEVELOPMENT OF PROJECTS AND PROGRAMS FOR DATA PROCESSING

[Abstract of article by Auerbach, D.]

[Text] A unified approach for developing data processing projects at the stages of design, implementation and testing is suggested and based on modern programming technology. The basis of this approach is the systematic decomposition of the task into a hierarchy of subtasks with well-defined rules of subordination for transfer of control, organization of the data stream and error processing. Brief information is given on the most important principles at each stage of development, implementation and testing. Current shortcomings in the method and ways of updating it are shown.

SOFTWARE FOR COMMUNICATION BETWEEN COMPUTER SYSTEM AND SPECIALIZED PROCESSORS FOR ARRAY PROCESSING

[Abstract of article by Ivanova, Y. and Atanasov, T.]

[Text] Features of software supporting methods of access to array processors are described. Two cases are considered: the YeS-2335 array processor without its own storage and the array processor with its own storage. The set of matrix operations supported is given.

UNIVERSAL DESCRIPTION LANGUAGE FOR CAD SYSTEMS

[Abstract of article by Botev, Kh. Ye.]

[Text] The language, developed for writing, inputting and updating data in a data file for a CAD system, features flexibility, universality and simplicity. A translator is provided. In each specific case, the user himself defines the structure and contents of the input data stream. Data on the file structure and rules for filling fields are stored in a permanent system file. The facilities developed may be used in any system with files of the type considered.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



ALBI-1 ALGORITHM STORAGE AND RETRIEVAL SYSTEM

[Abstract of article by Schubert, D.; Boelke, V.; and Ettrig, X.]

[Text] This system is based on the AIDOS programming system, which establishes a unified technique and criteria for algorithm description. The description language and elements of the retrieval language are discussed.

CONCEPT AND FEATURES OF PROGRAM PACKAGE FOR CAD OF DRILLING DEVICES

[Abstract of article by Rakovich, A. G. and Samek, A.]

[Text] Considered are the principles for design of a package and its main components with emphasis on describing the library of program modules for synthesis and documentaing designs. Information is given on hardware for implementation of the device design programs; a brief economic evaluation of the use of the package is given; information on prospects for its development and improvement is given.

CAD OF DIES

[Abstract of article by Pikler, D. and Turai, I.]

[Text] Generalized is the experience of developing a system for computer-aided design of dies, in which an attempt is made to solve the problem of limited knowledge by mechanical engineers of computers and the limited capabilities for writing algorithms for design procedures.

AUTOMATED SYSTEM FOR MASS PREVENTIVE TREATMENT OF POPULATION (SAMPO)

[Abstract of article by Astardzhiyan, G.; Bayev, B.; and Raychev, B.]

[Text] This system for preventive treatment of the population is an application program package to be used in real time under control of the SUIP program system, based on YeSTEL teleprocessing hardware. Hardware, package functional features and data bank organization are considered.

ACQUISITION AND PROCESSING SYSTEM FOR USSR GAS INDUSTRY DATA

[Abstract of article by Moletz, N.]

[Text] Results of special software development are given. General review of system structure and its components is given; areas of effective application are shown.

UNIFIED TECHNOLOGY FOR APPLICATION PROGRAM DEVELOPMENT

[Abstract of article by Baumbach, G.-D.]

[Text] Described is technological process of developing application programs. The technology is suitable for developing large program systems characterized by diverse functions, high complexity and dynamic development of area of application. Process begins with formulation of requirements for the program and is continued with design of the hierarchy of functions, data and documentation.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



ACQUISITION OF DATA ON COMPUTER HARDWARE OPERATION IN GDR

[Abstract of article by Hanisch, E.]

[Text] Experience gained in use of system for acquisition and analysis of data on computer and device reliability in GDR is described. The system effects continuous monitoring of the entire computer park and allows taking timely steps to maintain high reliability of manufactured hardware.

NEW SMALL COMPUTER HARDWARE FROM GDR

[Abstract of article by Leidler, K.]

[Text] Features, operational capabilities and brief information on software for two microcomputers, a printer and a mark reader in the SM family are presented.

MPL/600 ALGORITHMIC LANGUAGE

[Abstract of article by Aleksandrov, A; Yefremova, R.; and Aleksandrova, Z1.]

[Text] The MPL/600 algorithmic language oriented to the SM-600 microprocessor family and the MPL/600 cross-compiler for the Unified System of Computers are considered. Information on required computer configuration and average compilation time is given. Described are language capabilities and features, compiler operation, language syntax and rules for writing individual statements. Difference between MPL/600 and PL/1 is pointed out.

CONCEPT OF GENERALIZED (STANDARD) PROGRAM PRODUCTS

[Abstract of article by Yezhovich, E. and Vitkova, G.]

[Text] Described is the ADAPTOR program system, a formal means for creating and using flexible program products: standard elements of software for ASU's of varying scale and generality. The system belongs to the class of macroprocessors which do not depend on the type of source language.

COPYRIGHT: Izdatel'stvo "Finansy i statistika", 1981

8545
CSO: 8144/0138



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



TECHNICAL AND ECONOMIC JUSTIFICATION IN SELECTING CHARACTERISTICS OF PARAMETRIC SERIES OF COMPUTERS

Moscow VYCHISLITEL'NAYA TEKHNIKA SOTSIALISTICHESKIKH STRAN in Russian No 9, 1981 (signed to press 21 May 81) pp 5-11

[Article by Yu. P. Selivanov, engineer (USSR), and Yu. M. Novitskas, candidate of economic science (USSR)]

[Text] The principles, developed in machine building, for creating parametric series of machines are also applicable to computer science as the field concerned with the design, production and operation of data processing hardware. The use of these principles may be of considerable help in developing general approaches to the development of families of computers and in quantitative and qualitative evaluation of the characteristics of such a family.

In accordance with [1], as applied to computers, a parametric series should be understood as the aggregate of its elements--hardware, software and data compatible type sizes of computers for an operational purpose, i.e. intended for use in a specific macro area, having unified architecture, having limited interchangeability, differing from each other by the numeric value of the main parameter, throughput, and intended for meeting the given needs of society for processing of data.

As apparent from the above definition of a parametric series, throughput is usually accepted as the main computer parameter. On the one hand, this parameter is closely tied to all the other computer characteristics and often prescribes them, and on the other hand, it most fully reflects both the user properties of these machines and the cost for their production and operation.

In our view, based on the general theory for design of parametric series of machines, in the most general sense, throughput should be thought of as an indicator that takes into account not only the effective rate of the aggregate of functional units, but also the labor productivity of the personnel preparing the process of data processing and participating in it. This means that computer throughput must be spoken of as a man-machine system, which it basically is. Hence, it follows that the problem of increasing computer throughput must include the very important partial problem of reducing the share of cost of direct labor in the overall aggregate of direct and indirect labor needed to process a unit of information in solving one type problem. The way to solve this partial problem is to transfer to hardware and system software facilities for implementation in the automatic mode of those functions for solving the problem (development of the algorithm and program,



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



preparation of data, providing for an efficient mode of processing, operation of hardware, etc.) that have been previously performed manually in parctice. This leads not only to a decrease in total inputs of time needed to process a unit of information, but also to an increase in these inputs for the share of machine time at the expense of the manual, which is a mandatory prerequisite for further automation of all the processes in data processing in the national economy based on qualitatively new computer hardware.

This methodological approach to the evaluation of the throughput of a parametric series of computers permits determining the technical and economic effectiveness of the application of this parametric series in the entire macro area for which the series is designed. Also, this approach does not preclude the usual methods of evaluating throughput that permit determining the physical parameters of computers in solving problems of a specific type from the given macro area.

A major property that computers forming a parametric series must have is interchangeability. The classification by features of interchangeability of type sizes of a parametric series of machines and the characteristics of each type are given in [2]. The problem of interchangeability of computers as type sizes of a parametric series has not yet been adequately studied and requires separate consideration. By limited interchanegability, which computers in this series must have according to the definition of a parametric series given above, is meant the capability of replacing one type size of the series by at least one other, which, however, entails certain economic losses, for example, the substituting computer solves the same problems at the same qualitative level as that being replaced, but it costs more to do so.

A prerequisite and mandatory condition for interchangeability of computers forming a parametric series is their hardware, software and data compatibility.

Hardware compatibility of computers in a parametric series provides the capability of building them on the modular principle, i.e. by assembly of all computers in a parametric series from a unified set of functional devices, aggregates that are a set of structurally and functionally unified standard units and assemblies connected to each other by unified links. A relatively small number of unified units and assemblies is needed to obtain a large number of modifications of functional devices built on the modular principle. This approach reduces hardware development and manufacturing time and simplifies and reduces the cost of operating it. Unification of interfaces creates the capability of improving any of the devices independently of the others with subsequent inclusion of them into the configuration of a machine already in operation. This enables the development of the technical capabilities of the individual members of the parametric series, preventing obsolescence and extending thereby the life cycle of the series.

The modular principle permits developing the technical capabilities of an existing computer not only in depth, by improving the individual functional units of the machine, but also in width, by controlling the machine configuration. Using this principle of designing hardware combined with ensuring software and data compatibility and the use of special additional facilities permits forming complexes of computers, creating structures with a qualitatively new level of system organization and qualitatively new technical and economic parameters.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



Hardware, software and data compatibility of computers in a type size series is based on the unity of their architecture which has a modular structure and the development of which therefore can be effected without loss of software compatibility of the machines with computers previously manufactured.

A brief consideration of the characteristics of the properties of computers as type sizes of a parametric series permits formulating the requirements imposed on them from this viewpoint. They must be hardware, software and data compatible, have the property of interchangeability, have flexible configuration and lend themselves to complexing. These requirements, in particular, have been fully met by the second phase of the Unified System of Computers. In the first phase, the requirement of inetrchangeability was not fully realized since the YeS-1010 and YeS-1021 computers do not have the property of interchangeability with the other machines.

The features peculiar to a parametric series of computers and which characterize it as a type size series of machines are multidimensionality and multilevelness. Multidimensionality of a parametric series of computers is expressed in that the machines forming it differ from each other not only by the numeric values of the main parameter, but also by the numeric values of other parameters, basic and secondary, for example, by the main storage capacity, channel throughput, floor space occupied, etc. In some areas of application, these additional parameters may turn out to be the main ones. For example, for information systems with large external storage, channel throughput may be the decisive parameter in choosing a machine.

The multilevel nature of a parametric series of computers is manifested in the fact that not only the computers, but also the functional units forming them and the modules of the lower levels, including the integrated circuits, are elements of specific parametric series. In other words, each computer, being an element of a parametric series of these machines, consists of devices, units and elements, etc. that in turn are elements of a corresponding partial series. The multilevel property is especially characteristic of the second phase of the Unified System of Computers. In the first phase, only central units formed parametric series. One could single out groups of peripherals that did not form a parametric series for the main parameter, for example, several types of disk storage units with identical capacity of 7.25M bytes.

A major characteristic of a parametric series of computers, just as of any other parametric series, is its density, i.e. number of type sizes. Parametric series of computers capable of performing the same amount of computational operations at the same qualitative level may consist of a diverse number of type sizes (differ in density), and series with an identical number of type sizes (equal in density) may consist of type sizes diverse in the numeric value of the main parameter.

Thus, a certain amount of work on data processing may be performed with the aid of different parametric series of computers. One cannot compare computers in a parametric series with separate computers that do not form this series, since the parameters of the computers forming the parametric series are selected and optimized on the basis of the tasks set for the entire series. The user cost of data processing in all these alternatives is identical since they permit performance of the same amount of work at the same qualitative level, but the labor cost is not equivalent



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



since the performance of the same amount of work requires different socially needed inputs of labor. In other words, inputs for development, production, commissioning and operation of computers depend on how the series is constructed (on the number of type sizes and density of their arrangement in its individual intervals, defined by the numeric value of throughput peculiar to the models in this series).

Consequently, the problem of choosing the optimum parametric series of computers from all possible alternatives has to be solved. To solve this problem, it is expedient to use the method of comparative economic effectiveness, the calaculation of indicators of which is based on monetary measurement of inputs. The inputs listed above associated with the development of parametric series of computers are synthesized in two monetary indicators: capital investment and operating expenses.

Capital investment associated with development of the i-th alternative of a parametric series may be defined by the formula

$$K_{ps\,i} = \sum_{j=1}^{m} K_{COM\,ij}, \qquad (1)$$

where $K_{COM\ ij}$ is the capital investment needed for development, manufacture and commissioning of a computer of the j-th model of the i-th parametric series, rubles/computer.

The value of $K_{COM\ ij}$ for each j-th model of computers in the series has to be determined with regard to the wholesale price for the minimum configuration of the model; inputs for design of the parametric series, including inputs for development of the software system per model (if for some reason these costs are not included in the wholesale price for the machine); inputs for transportation of the computer to the installation site, assembly, adjustment and startup of it; and costs of machine floor space occupied (including space needed for servicing and repair).

Expenses for operation of each i-th alternative of the design of a parametric series of computers can be calculated as

$$E_{ps\,i} = \sum_{j=1}^{m} E_{COM\,ij}, \qquad (2)$$

where $E_{COM\ ij}$ is the annual expense for operating a computer of the j-th model of the i-th parametric series, $\frac{\text{rubles/year}}{\text{computer}}$.

The value of $E_{COM\ ij}$ must include: basic and supplementary salaries for personnel servicing the type configuration of the machine; amortization of the computer; cost of services performed by specialized enterprises for complex centralized maintenance of computer hardware per machine; cost of routine computer repair performed by maintenance personnel; cost of power consumed; cost of auxiliary materials (magnetic tapes, perforated tapes, paper, etc.); and amortization of machine floor space occupied.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



The economically optimal alternative of a parametric series of computers is chosen based on the monetary criterion of comparative effectiveness, the indicator of unit adjusted costs for the i-th alternative of a design of the series. This indicator is determined by the expression

$$W_{d\,ji}=\frac{S_{ps\,i}+\varepsilon_n K_{ps\,i}}{q_{ps\,i}}=\frac{\sum_{j=1}^{m} E_{COM\,ij}+\varepsilon_n\sum_{j=1}^{m} K_{COM\,ij}}{\sum_{j=1}^{m} q_{COM\,ij}}, \qquad (3)$$

where $\varepsilon_n$ is the normative factor of effectiveness of additional capital investment, 1/year;

$q_{ps\,i}$ is the total annual throughput of all models of the computers forming the i-th parametric series, $\frac{\text{thousands of operations/year}}{\text{series of computers}}$;

and $q_{COM\,ij}$ is the annual throughput of a computer of the j-th model of the i-th parametric series, $\frac{\text{thousands of operations/year}}{\text{computer}}$ $\left(\frac{\text{type tasks/year}}{\text{computer}}\right.$ etc.$\left.\right)$

Considered economically optimal is that i-th alternative of the design of the parametric series of computers by which

$$W_{d\,ji} = \min. \qquad (4)$$

Thus, based on what has been said above and with regard to [1], an economically optimal parametric series of computers should be considered a parametric series of these machines with such numeric values of the main parameter, throughput and other (basic and secondary) parameters with which the specified national economic needs for data processing are met with the least unit adjusted costs.

The basic requirements imposed on an economically optimal parametric series of computers are:

long-term nature, ensured by including in the series future type sizes of computers, the need for which cannot be met immediately or may arise later;

optimality of period of manufacture, i.e. an economically optimal term for the series models being in production, determined by the technical level and quality of these models, rate of technical progress in computer manufacture and sectors related to it, and rate of change in the national economy of the volume and structure of information subject to processing on computers of this type;

technical and economic justification which includes a technical and economic analysis of the parameters of computers manufactured by domestic and foreign industry; selection of computer parameters determining the technical level and quality of these machines; analysis of the actual data and forecasts for the volumes and structure of information subject to processing on computers of a given type; definition of the need by the national economy for such machines; selection of a rational and progressive base design for the computers and the software system (from the viewpoint of performing computations with the minimal socially needed inputs of labor); determining and comparing cost required to implement the different alternatives of a parametric series of computers of a given type, etc.;

optimality of nomenclature of computer technical parameters prescribed by standards, and selection based on objective criteria not only of the main, but also of the basic parameters of the machines from the set of existing parameters.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



In setting out to design a parametric series of computers, first of all one must establish its users, the most typical operating conditions and the demand by the national economy for the machines of the various type sizes. After this, the alternatives of the parametric series have to be outlined and the alternative that most fully meets the requirements of production adn operation of these machines has to be chosen on the basis of technical and economic analysis and with regard to the criterion of optimality (4).

The starting point for development of an economically optimal parametric series of computers is the definition of demand by the national economy for the given type of machines. In turn, computer demand is based on the data volume and structure that it is economically expedient to process by using machines of a given type. Great difficulties arise in analyzing the demand for tasks to be handled by using computers and the distribution of these tasks by parameters describing them. Still, this is the only kind of analysis that permits defining the demand by the national economy for computers of specific type sizes with technical parameters optimally corresponding to parameters of the tasks to be handled.

BIBLIOGRAPHY

1. "Tipovaya metodika optimizatsii mnogomernykh parametricheskikh ryadov" [Standard Technique for Optimization of Multidimensional Parametric Series], Moscow, Izdatel'stvo standardtov, 1975.

2. Ivanov, A. V., "Ekonomika ryadov mashin" [Machine Series Economics], Moscow, Izdatel'stvo standartov, 1975.

COPYRIGHT: Izdatel'stvo "Finansy i statistika", 1981

8545
CSO: 8144/0138



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



CONCEPT OF YES-1060 COMPUTER DESIGN AND DEVELOPMENT

Moscow VYCHISLITEL'NAYA TEKHNIKA SOTSIALISTICHESKIKH STRAN in Russian No 9, 1981 (signed to press 21 May 81) pp 12-19

[Article by V. S. Antonov, candidate of engineering science (USSR), K. S. Orayevskiy, engineer (USSR), and B. B. Avtonomov, engineer (USSR)]

[Text] A trend in the evolution of the Unified System of Computers is the improvement of models previously produced by extending their functional capabilities. Such development assumes maximum preservation of the basic structure and conceptual solutions adopted in the central units of the computer, the processor, the IO channels and the main storage unit, and of their interface principles. This permits extending the term for series production of the computer and at the same time provides the capability of developing for the computer newly introduced functions, principles for implementation of them and new versions of the operating systems to facilitate their introduction and ensure continuity in the models being produced. This approach to computer development as a whole becomes especially important in developing the processor, since introduction of new functions, primarily expanding the instruction set, introducing new types of interruptions, etc. is reflected mainly in the processor.

Now, when the program for further development of the Unified System of Computers has been formulated, it is useful to generalize the experience of developing the YeS-1060 computer.

In designing the YeS-1060 computer, the following tasks were solved:
development of a high-throughput computer that implements the principles of operation of the second phase of the Unfied System of Computers;
raising the technical and economic characteristics compared to those of the computers in the first phase of the Unified System;
raising the internal speed of the processor;
increasing the capacity of main storage;
raising operating reliability;
development of checking and diagnostic facilities; and
introduction of the new technological design and element base.

Development of the YeS-1060 computer began while the principles of operation for the second phase of the Unified System of Computers were still being worked out, but the basic concepts had already been formulated and adopted, such as the organization and facilities for providing virtual storage, expanding the instruction set and the



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



system of interruptions. The YeS-1060 computer had to become a transition model; therefore, it was oriented to further development. In doing so, a path of stage-by-stage development was chosen, and the computer of the first stage had to become a series model. Let us consider the ways and emthods of solving the basic problems in designing the YeS-1060.

The main technical and economic characteristic of a computer is the throughput-cost ratio. This characteristic can be improved by either increasing computer throughput or reducing hardware cost. In turn, throughput can be increased through both more efficient organization of the computing process, raising the level of multiprogramming and reducing unproductive inputs of machine time, and through increasing the internal speed of the processor, IO channel throughput and rate of operation of the external devices. However, increasing processor speed and IO channel throughput requires additional inputs of hardware, which increases computer cost somewhat. Increasing computer throughput requires in the first place expanding the capacity of main storage which enables efficient use of the computer.

In the YeS-1060 computer, the technical and economic characteristics have been improved as a result of the optimal combination of the methods mentioned. Thus, for example, considering the capabilities of teh element base used, a relatively simple structure was selected for the processor with overlap of instruction processing - three levels and application of efficient microprogram control, which permitted o.- taining a throughput of over one million operations per second with comparatively low costs for hardware.

A large main storage unit was developed in the YeS-1060 computer in the first stage through the incorporation of more modern design solutions for the ferrite storage used previously. This enabled quadrupling the storage capacity in a rack. In the second stage, integrated circuits were used as the storage medium.

A major problem in developing a computer is raising operating reliability, which means faultless operation, automatic recovery of the computing process when system serviceability is damaged and reducing repair and maintenance time. The checking and diagnostic facilities are aimed at accomplishing this.

Parity checking facilities incorporated in the YeS-1060 computer cover up to 90 percent of the equipment in the main processing assemblies and data transmission paths. Hamming code checking was also introduced for main storage. This enables detection of single and double errors and correction of single errors of data in storage.

Diagnostic facilities include both software and microdiagnostic tests that together with the checking facilities permit detection of practically all malfunctions or failures of central unit hardware, processor, storage and channels, and localization of malfunctions with a precision down to a group of TEZ's [standard exchange cards].

These facilities reduce considerably computer maintenance and repair time and also provide for operation of recovery facilities. The latter include hardware retry of instructions and procedures in the CPU and channels and software facilities in the operating system that provide for retry of programs or sections of programs.

Standard solutions for the Unified System of Computers were adopted as the basis for mechanical processing of the YeS-1060 computer. To raise the level of machine



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



manufactureability in series production and improve operating qualities, a number of new industrial processes was introduced in machines of this class, such as wire-wrapping in assembly of panels, assembly of inter-panel connections by flat cable, etc.

ECL, the IS500 series of inetgrated circuits, was chosen as the computer element base. This series includes 26 types of circuits with medium density of about 10 gates per package with typical delay time of 1.5-2 ns per element and average dissipated power of about 250-300 mW per package. Subsequently, the number of types of integrated circuits reached 47 as a result of introduction of more complex functional elements and development of storage elements. Average number of gates per package grew to 15-20.

Using the IS500 made it possible to implement an advanced structure for the central units in the computer with rather limited structural bulk and with a relatively short operating cycle.

The Processor. In analyzing the capabilities of improving the processor structure, the conclusion was drawn that units such as the instruction processing unit and the arithmetic unit control unit must be the most dynamic and adaptable. Analysis of the experience in developing and adjusting the YeS-2060 processor showed that the greatest difficulties are encountered in developing the control circuits.

The primary use of microprogram control and the use of loadable storage of microprograms allow for development of the processor, introduction of new functions and extension of the instruction set without substantial adjustment to the apparatus. Microprogram control also provides efficient diagnostic facilities that can be developed without additional costs for equipment. The expediency of this solution was confirmed both during adjustment of the pilot model and during development of new versions of the operating systems.

The YeS-2060 processor (fig. 1) has three basic units: the storage control (UP) unit, the central control and instruction processing (TsU) unit and the arithmetic unit (BA). The storage control unit services requests coming from the processor units and IO channels for reference to main storage. The main functional assemblies in the storage control unit are high-speed buffer storage (BP), IO channel buffer storage (BK) and dynamic address translation facility (DTA) and the storage adapters (AP). The central control and instruction processing unit fetches instructions from main storage, processes them in the overlap mode, generates operand addresses and fetches operands from local and main storage. Its functional assemblies include the buffer registers for instruction processing (RB), the instruction counter (SchK) and the microprogram control unit (MU). The arithmetic unit performs operations on operands in the process of executing operations and writes the results to local and main storage. The main functional processing assemblies in the arithmetic unit are the parallel adder (SM), the shifter (SD) and the fast multiplier (UM). The control units for the central control and instruction processing unit and for the arithmetic unit are microprogrammed. Control signals are generated by the microprogram control units.

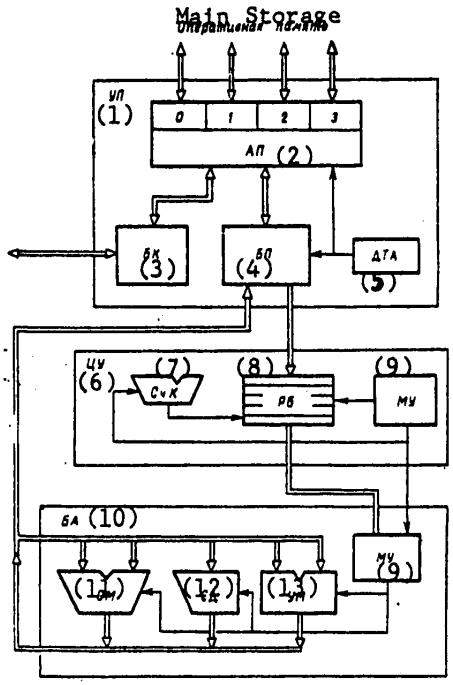
The chosen structure of the processor permits solving the problem of matching the relatively long access time from main storage with the short CPU cycle and provides for efficient overlap of the operation of the IO channels and CPU with main storage.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6






Key:

1. UP [storage control unit]
2. AP [storage adapter]
3. BK [IO channel buffer storage]
4. BP [high-speed buffer storage]
5. DTA [dynamic address translation]
6. TsU [central control and instruction processing unit]
7. SchK [instruction counter]
8. RB [buffer registers for instruction processing]
9. MU [microprogram control unit]
10. BA [arithmetic unit]
11. SM [parallel adder]
12. SD [shifter]
13. UM [fast multiplier]

Fig. 1. Structural diagram of the YeS-2060 processor

These problems are solved by incorporation of the mode of interleaving of main storage (interlacing of addresses), the high-speed buffer storage and the buffer storage for the IO channels in the storage control unit.

The specified speed of the CPU is achieved through, first, overlapped processing of instructions which permits obtaining results of execution every two steps of the processor operation; second, more efficient structure and algorithms for execution of operations in the arithmetic unit which decreases the average number of steps in instruction execution; and third, as a result of incorporation of the high-speed buffer storage which reduces the effective cycle of main storage threefold.

CPU characteristics are:

| | | |
|---|---|---|
| principle of control | | hardware-microprogram |
| instructions realizable | | 183 |
| overlap levels | | 3 |
| throughput, measured on the basis of statistical mixes for scientific and technical tasks | | 1,040,000 instructions/second |
| operating step | | 160 ns |
| buffer storage: | capacity | 8K bytes |
| | cycle | 160 ns |

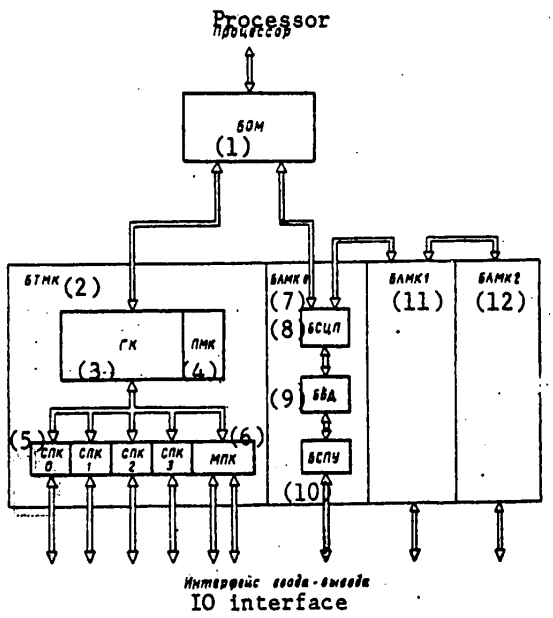


APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



The YeS-4001 Universal Multiplexer Channel extends the functional capabilities of the computer IO system. It unites groups of functionally independent channels in one device (fig. 2). The byte-multiplexer channel (BTMK) services low-speed




Key:
1. BOM [unit of exchange buses]
2. BTMK [byte-multiplexer channel]
3. GK [main channel]
4. PMK [local storage for main channel]
5. SPK (0, 1, 2, 3) [selector subchannels]
6. MPK [block-multiplexer subchannel]
7. BLMK0 [block-multiplexer channel]
8. BSTsP [CPU interface unit]
9. BBD [data buffer unit]
10. BSPU [peripheral interface unit]
11. BLMK1 [block-multiplexer channel]
12. BLMK2 [block-multiplexer channel]

Fig. 2. Structural diagram of the YeS-4001 universal multiplexer channel

peripherals in both the multiplexer and burst modes, and medium-speed devices in the burst mode. The basic functional units of the byte-multiplexer channel are the main channel unit (GK), four selector subchannels (SPK) and the block-multiplexer subchannel (MPK). The main channel unit enables exchange of signals with the CPU and storage control unit, exchange of data and control words with main storage, and storage and modification of control data. The main channel exchanges data with the CPU through the unit of exchange buses (BOM). The main channel exchanges data with the multiplexer subchannel in bytes. Storage and unpacking of double words is performed in local storage (PMK) for the main channel. Selector subchannels have their own data buffers and exchange data in double words with the main channel.

A block-multiplexer channel services high-speed peripherals. The device is connected through a fast two-byte interface, permitting data transfer in both the one-byte and two-byte modes. The basic functional units are the CPU interface unit (BSTsP), the peripheral interface unit (BSPU) and the data buffer unit (BBD). Data is exchanged with the CPU through the unit of exchange busses in double words over a bus common to the block-multiplexer channels.

The block-multiplexing mode allows a considerable increase in IO system efficiency and efficiency in the use of channel equipment. The universal channel supports: the indirect addressing mode in the channels;



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



the two-byte interface with throughput of 3M bytes/second for connection of advanced high-speed devices;
expansion of the IO instruction set; and
facilities for instruction retry and recovery after failures.

Technical characteristics for the YeS-4001 channel are:

| | |
|---|---|
| block-multiplexer channel: | |
| number | 3 |
| throughput, one-byte interface | 1.5M bytes/second |
| throughput, two-byte interface | 3M bytes/second |
| byte-multiplexer channel: | 1 |
| selector subchannels | 4 |
| block-multiplexer subchannels | 192 |
| selector subchannel throughput | 0.5M bytes/second |
| block-multiplexer subchannel throughput | 0.1M bytes/second |

The main storage unit for the YeS-1060 computer is built with integrated circuits with a capacity of 16K bits per package, giving 8M bytes in the standard rack, which also holds the power supply for main storage. The use of dynamic integrated circuits required the regeneration mode in the CPU storage control unit.

Unit cycle time is 680 ns and access time is 520 ns.

In the first stage of development of the YeS-1060 computer, basic efforts were directed toward developing a processor as a unit reflecting to the highest extent the the technical characteristics of the computer.

The YeS-4012 and YeS-4035 units, that were part of the YeS-1050 computer, were used as the IO channels. The YeS-3206 main storage unit was built with ferrite cores with the application of technology developed in the main storage unit for the YeS-1050. The power supply system also made use of the secondary power supply units for the YeS-1050 computer. This permitted using series manufactured devices and units and thereby reducing the labor-intensiveness of development and the period for putting the computer into series production.

The standard configuration of the central units in the computer in the first stage was: the YeS-2060 CPU, the YeS-4012 multiplexer channel, two YeS-4035 selector channels, and two YeS-3206 main storage units with a capacity of 2M bytes.

The standard configuration took up nine racks, including four for the CPU and channel power supply system.

In the second stage of development of the YeS-1060, the problem of developing a computer that fully implements the principles of operation of the second phase of the Unified System of Computers was solved; this required development of the new universal channel, the YeS-4001. The problems of developing main storage with integrated circuits and improving the power supply system by using more powerful secondary power supplies were also solved. The main efforts in this stage were aimed at developing the YeS-4001 universal channel and developing operating system versions for the virtual organization of storage.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



In the second stage, in the standard configuration, the number of IO channels was also increased to four and main storage capacity to 8M bytes, and the operating, technical and economic characteristics of the model were substantially improved.

The standard configuration of central units in the YeS-1060 computer in the second stage includes the YeS-2060 CPU, the YeS-4001 unit with one byte-multiplexer and three block-multiplexer channels, and the 8M-byte main storage unit.

The total racks for the central units is four.

During the period of improving the first-stage computer for series production, CPU facilities were developed that support the principles of operation of the second phase of the Unified System of Computers, the OS 6.1 version of the operating system and facilities for recovery after failures in the CPU.

The experience of the development, adjustment and operation of the YeS-1060 computer confirmed the correctness of the chosen direction of staged development of the model, which permitted development of a computer with high operating, technical and economic characteristics. Further improvement of the YeS-1060 may involve reducing the CPU operation step, raising the reliability and expanding the diagnostic facilities for maintenance.

COPYRIGHT: Izdatel'stvo "Finansy i statistika", 1981

8545
CSO: 8144/0138



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



UNIFIED SECONDARY POWER SUPPLIES FOR SERIES-2 UNIFIED SYSTEM COMPUTERS

Moscow VYCHISLITEL'NAYA TEKHNIKA SOTSIALISTICHESKIKH STRAN in Russian No 9, 1981 (signed to press 21 May 81) pp 19-28

[Article by B. N. Fayzulayev, doctor of engineering science (USSR), Ye. I. Batyukov, engineer (USSR), and Zh. A. Mkrtchyan, engineer (USSR)]

[Text] Distinctive features of the second phase of the Unified System of Computers are the wide range of throughput (from tens of thousands to several millions of operations per second), improvement of the throughput-cost ratio, use of fast integrated circuits with an enhanced degree of integration, unified principles for the design of power supplies and the use of unified secondary power supplies (UVIP).

The power supply system (SEP) for the Unified System Computer hardware is part of the electronic equipment and largely determines the technical and economic indicators for computers in the Unified System and their individual units; therefore, improving the parameters of the power supply system was a major task in developing the Unified System of Computers.

The power supply system for the Unified System of Computers is the aggregate of the transformers for the industrial power supply line, distributing, regulating and switching devices, protection and control units, and the secondary power supplies that form the main part of the power equipment for the hardware.

The basic requirements imposed on hardware power supply systems for the Unified System of Computers are:

modular principle of design;
structural and electromagnetic compatibility between the power supplies and electronic and logic units, allowing them to be used together in the basic design;
unification and standardization of the nomenclature and design of secondary power supplies (VIP) [SPS] and power control units (BUP);
unified system interface for power supplies;
high stability of voltages output from SPS;
enhanced immunity to the effect of main power line and cross noise;
provision of the necessary protection and control functions with stable thresholds for operation;
high efficiency factor; and
high degree of unification of circuit engineering and design solutions with regard to reducing labor-intensiveness of development of various systems and organization of their large series production.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



These requirements were worked out at the stage of developing the first phase of the Unified System of Computers and formulated in a series of standards and methodological instructions.

During development of the units and systems for power supply to the "Series-1" hardware, much attention was paid to solving the problems of electromagnetic compatibility between the SPS and electronic equipment and to improving SPS unit characteristics. Let us discuss these problems in more detail.

Computer power supplies for the first and second generations were largely based on line regulators that are characterized by a low level of pulse and high-frequency noise; their unit indicators fully meet the requirements of the electronic equipment built with discrete elements. With the shift from discrete components to integrated elements in third-generation computers, the absolute noise immunity of the element base declined due to the decrease in amplitudes of the working signals, while the SPS noise level increased considerably due to the use of pulse regulation methods, the use of key SPS and increased power consumption per structural unit.

Due to the small working signals of modern integrated circuits [IC] and the small absolute value of noise immunity, it became necessary to protect computer hardware from noise in the industrial power line and cross noise generated by the computer hardware itself. The shape and amplitude of external noise depends on the nature of the occurrence. However, most dangerous is industrial noise that occurs on the main power line. Recording and studying it is difficult because it is irregular and has a wide range in duration of effect and amplitude. Reliable operation of hardware, in particular the time between breakdown and failure, depends on the amount and spectrum of noise affecting the power supply and the capability of the latter to impede the passage of noise to the load to a considerable extent. From the viewpoint of immunity to industrial and cross noise, the following requirements on the Unified Computer System power supply system were formulated as a result of research:

immunity to the effect of voltage deviations in the industrial supply by 80V for 20-30 ms; and
immunity to the effect of voltage spikes with an amplitude of 1000-1500V and duration of 2-10 microseconds.

Electromagnetic compatibility under the conditions of main supply, cross and internal noise was provided for the SPS as a result of adopting a number of complex measures:

exacting design of the electronic portion of the power supply system and system for electronic assembly of hardware elements and assemblies;
use of main line noise suppression filters in the SPS and devices;
use of converters for the primary supply line (motor-generators, static converters, etc.); and
development of a protection system for units and devices.

In first-phase Unified System computer devices, the power supply system makes up a significant portion of all electronic equipment (50 to 70 percent of computer bulk). This is because the size of electronic equipment was considerably reduced compared to that in second-generation computers owing to the application of IC's, while SPS consisting of various types of discrete electronic and radio items do not lend



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



themselves to effective microminiaturization. Because of this, for third-generation computers, reducing the structural bulk of power supplies and in the first place increasing the unit indicators of SPS became a basic problem. The increased requirements on SPS parameters and device packaging density, and the need for raising the manufactureability and reducing the labor intensiveness in manufacturing power supplies required substantial improvement in power supply characteristics at the stage of developing the second phase of the Unified System.

At the same time, the structure of the power supply system adopted for the Unified System had notable advantages: standard interface and modular principle of design based on the developed unified series of unified power supplies (UBP) permitted development and putting into production power supply systems for all hardware within a short term.

The design and introduction into large series production of unified power supplies indicated that that was the only way to enable development of the required quantity of SPS, install them in the various Unified System hardware and achieve high quality with minimal outlays.

The modular structure of the functionally complete units with an autonomous system for protection and control is a characteristic feature of the design for power supply systems for the Unified System of Computers. Use of it made it possible to:

organize large series production of the units used in all Unified System hardware;
enable centralized development and high quality fo the units;
employ joint packaging of power supplies and electronic units in the unified base design for various types of hardware;
reduce the labor intensiveness for development of Unified System hardware; and
continually improve and modernize power supplies irrespective of the development and production of Unified System hardware.

Considering the advantages of the power supply system used in the first phase of the Unified System of Computers, listed above, this structure was kept when the power supply systems and units were developed for "Series-2" hardware.

The main task in developing power supplies for second-phase Unified System hardware was to reduce the size of the power supply system and increase specific output power of SPS two-three fold compared to that for the first-phase unified power supplies, noise immunity and efficiency factor. The latter is necessary primarily to ensure normal heat conditions under the conditions of increasing specific output power while keeping the previous structures and to save energy while power consumption is increasing continually in the national economy.

A unified series of unified secondary power supplies was developed and introduced into series production for second-phase Unified System hardware. These units meet the requirements formulated above and are built on the basis of high-voltage key regulators on the high-voltage side with application of high-frequency power semiconductor instruments manufactured domestically.

The unified nomenclature of the unified secondary power supplies and the modular compatibility of them with the basic designs of all Unified System hardware have enabled their wide application in devices used for various purposes.

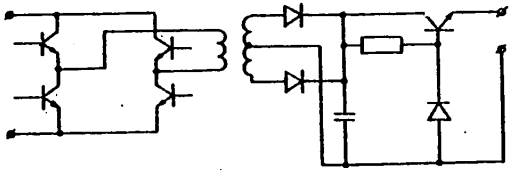
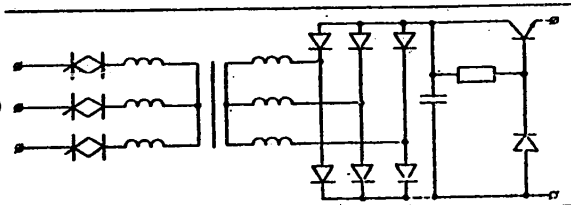
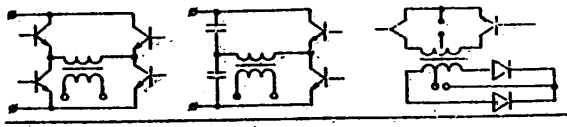


APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

FOR OFFICIAL USE ONLY

Table 1. Applicability of different circuits for combined conversion and regulation of electricity in secondary power supplies for REA [radioelectronic apparatus] and computer hardware (1 - applied; 0 - not applied)

| Conversion and regulation principle | Circuit | USSR | | USA | | UK | | FRG | | It. | | Jap. | | Fr. | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | СССР | | США | | Англия | | ФРГ | | Италия | | Япония | | Франция | |
| | | ЭВМ | РЭА | ЭВМ | РЭА | ЭВМ | РЭА | ЭВМ | РЭА | ЭВМ | РЭА | ЭВМ | РЭА | ЭВМ | РЭА |
| | | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| Variable transistorized inverter with output line regulator |  | 1 | 1 | 1 | 1 | | 1 | 1 | 1 | | | 1 | 1 | | |
| Controlled rectifier (based on phase regulator) with output line regulator |  | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | | 1 |
| TRANSFORMERLESS INPUT Variable inverter with transistors with pulse-width modulation (ShIM) |  | 1 | 1 | 1 | 1 | | 1 | 1 | 1 | | | 1 | 1 | | 1 |
| Single-cycle transistorized inverter with pulse-width modulation |  | 1 | 1 | 1 | 1 | | 1 | | 1 | | | 1 | 1 | | 1 |

Key: It. = Italy
Jap. = Japan
Fr. = France

odd numbered columns indicate whether applied for computer hardware, even numbered for radioelectronic apparatus



FOR OFFICIAL USE ONLY

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



[Continuation of Table 1]

| | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Variable transistor-thyristor inverter with pulse-amplitude modulation (AIM) | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Variable thyristor inverter | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| CONVERTERS WITH TRANSFORMER INPUT | | | | | | | | | | | | | | |
| with use of resonant S-circuits | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| with use of magnetic amplifiers (MA) | 0 | 1 | 1 | 1 | 1 | | | | | | 1 | 1 | 0 | |
| with phase regulation in AC circuit | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | | 1 |
| with use of parametric transformer | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | | | | | 0 | 0 |
| with use of ferroresonant regulator | | 1 | | 1 | | 1 | | 1 | | 1 | | 1 | | 1 |





APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



Table 2. List of unified secondary power supplies (UVIP)

| (1) Тип УВИП | (2) Шифр | (3) Габаритная доля типовой панели ЕС ЭВМ | (4) Масса, кг | (5) Потребляемая мощность, Вт | (6) Питающая сеть, число фаз | (7) Удельная мощность (8) Вт/дм³ | (9) Вт/кГ |
|---|---|---|---|---|---|---|---|
| 2B 15A | EC-T200/B007 | 1/8 | 5 | 90 | 1 | 9,0 | 6,0 |
| 2B 45A | EC-T200/B005 | 1/4 | 8 | 200 | 3 | 12,6 | 11,2 |
| 2B 100A | EC-T200/B016 | 1/2 | 18 | 600 | 3 | 13,5 | 11,1 |
| 2,4B 8A | EC-T200/B019 | 1/16 | 1,5 | 50 | 1 | 11,2 | 12,8 |
| 5B 8A | EC-T200/B018 | 1/16 | 2,5 | 80 | 1 | 23,5 | 16,0 |
| 2,4B 80A | EC-T200/B015 | 1/2 | 18 | 600 | 3 | 13,0 | 10,7 |
| 5B 15A | EC-T200/B022 | 1/8 | 5 | 95 | 1 | 22,7 | 9,0 |
| 5B 40A | EC-T200/B004 | 1/4 | 8 | 350 | 3 | 28,1 | 25,0 |
| 5B 90A | EC-T200/B002 | 1/2 | 11,5 | 700 | 3 | 30,4 | 39,0 |
| 5,2B 15A | EC-T200/B006 | 1/8 | 5 | 140 | 1 | 23,6 | 15,6 |
| 5,2B 40A | EC-T200/B003 | 1/4 | 8 | 350 | 3 | 29,3 | 26,0 |
| 5,2B 80A | EC-T200/B001 | 1/2 | 11,5 | 700 | 3 | 25,4 | 36,0 |
| 6,3B 5A | EC-T200/B017 | 1/16 | 1,5 | 60 | 1 | 18,5 | 17,5 |
| 12B 10A | EC-T200/H002 | 1/4 | 8 | 170 | 1 | 16,9 | 15 |
| 12,6B 2A | EC-T200/B012 | 1/16 | 1,8 | 60 | 1 | 14,8 | 14,0 |
| 12,6B 20A | EC-T200/B013 | 1/2 | 10 | 450 | 3 | 17,0 | 25,2 |
| 15B 6A | EC-T200/H001 | 1/4 | 8 | 170 | 1 | 12,6 | 11,0 |
| 20B 2A | EC-T200/B011 | 1/16 | 1,8 | 60 | 1 | 23,5 | 22,2 |
| 20B 4A | EC-T200/B024 | 1/8 | 5 | 125 | 1 | 24,2 | 16,0 |
| 20B 15A | EC-T200/B009 | 1/2 | 20 | 600 | 3 | 20,2 | 15,0 |
| 27B 2A | EC-T200/B010 | 1/16 | 1,5 | 80 | 1 | 31,7 | 30,0 |
| 27B 2A | EC-T200/B010 | 1/16 | 1,8 | 80 | 1 | 31,7 | 30,0 |
| 27B 3A | EC-T200/B025 | 1/8 | 5 | 125 | 1 | 24,5 | 16,0 |
| 27B 13A | EC-T200/B008 | 1/2 | 20 | 600 | 3 | 23,7 | 17,5 |
| 48B 8A | EC-T200/H004 | 1/2 | 11,5 | 50 | 1 | 26,0 | 33,3 |
| 5B 110A | EC-1052/H002 | 1/2 | — | — | 3 | 40,0 | 40,0 |

(10) Габаритные размеры долей типовой панели ЕС ЭВМ, мм:
1/2—386×178×215;
1/4 — 179×178×215;
1/8 — 89×178×215;
1/16 — 44×178×215.

Key:
1. power supply type, B = Volts, A = Amps, (, = .)
2. model number, EC = YeS, T = T, B = V, H = N
3. dimensional share of standard Unified System panel
4. weight, kg (, = .)
5. power consumption, watts
6. feed line, number of phases
7. specific power
8. $W/dm^3$ (, = .)
9. W/kgf (, = .)
10. dimensions of share of standard Unified System panel, mm





APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



There is a trend to using SPS with transformerless input in both domestic and foreign development of computer hardware power supplies. Table 1 shows the applicability of the different principles for electrical conversion and regulation in SPS for computer hardware and other radioelectronic apparatus. Analysis of scientific, technical and patent literature indicates that the majority of foreign firms tend to use the method of pulse-width voltage regulation; in the process, used as output stages are transistor bridge and half-bridge variable inverters with transformerless input. In a number of cases, single-cycle converters and line voltage regulators have been found to be more efficient for SPS with output power of less than 100W.

Table 2 lists the unified secondary power supplies developed and put into large series production. Characteristics of the unified power supplies and unified secondary power supplies [SPS] are:

| | UPS | SPS |
|---|---|---|
| Total instability, % | 5 | 3±4 |
| Retention of output voltage, microsec. | 0.1±1 | 10±30 |
| Efficiency factor, % | 35±50 | 65 |
| Specific output power, $W/dm^3$ | 10÷12 | 30±40 |
| $\frac{\text{R dissipation}}{\text{R load}}$ | 1.0±1.85 | 0.55 |

From this information, it can be seen that with the design of the SPS, specific output power was increased two to threefold, amount of total instability of output voltage was reduced 1.5-fold and the efficiency factor was raised 1.3-2 fold which permitted keeping the total heat release within the supply casing.

The basic indicators of the key SPS with transformerless input, used in the central part of the computers, in specific indicators are similar to the SPS of the socialist countries.

It is interesting to compare the basic specific indicators for the power supply systems and units for the first and second phases of Unified System hardware. The first-phase system was built on the base of unified power supplies (UBP) with transformer input, developed by using linear and combined principles for converting the electricity. The most powerful unit in this series, the UBP 5V/60A, was designed according to the structural scheme with a variable thyristor inverter with transformerless input and 1 kHz conversion frequency. Introduction in Unified System power supply systems of UBP series units afforded their electrical and structural compatibility with Unified System hardware; in the process, the ratio of power supply size to total computer device size was 0.5 to 0.7.

Using modern methods of high-frequency conversion and pulse regulation of electricity in power supplies for the second phase of the Unified System of Computers permitted reducing the size taken up by the power supply to 0.3-0.5 of the total size of the electronic equipment. Thus, for example, the power supply for the YeS-1060 computer processor with UBP type units (thyristor inverter with 1 kHz conversion frequency) took up 2.5 standard racks; switching to the SPS permitted placing the corresponding power supply system in just one rack. Thus, the change from UBP to the more economical and smaller SPS with transformerless input and conversion of electricity at the increased switching frequency of 20 kHz permitted reducing the



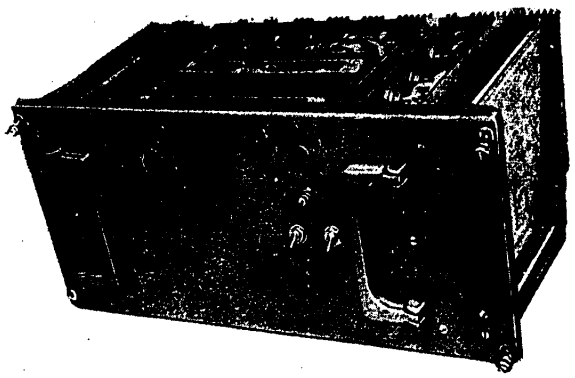
APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



corresponding size of the power supply system two to threefold. This made it possible to place the electronic portion of the hardware and power supply units in one cabinet and to meet the requirement on electromagnetic and structural compatibility between the power supplies and Unified System computer hardware. The increased requirements for electromagnetic compatibility were met, including through introduction of motor-generators in the computers.

Structurally, UVIP [unified secondary power supplies] are units (see photo) with a width equal to a main panel. The height and depth of all units are identical. The maximum dimensions of these SPS correspond to the dimensions of a half panel, and the minimum to 1/16 of a panel. Placed on the front panel are controls for the continuously adjustable level of output voltage, for switching to the mode of preventive change in the levels of output voltage by $\pm$5% of the rated value, and a warning light. The unit rear panel has a connection for the control and protection circuits, a connection for input of the main supply voltage, and an output for the feedback and output voltage circuits for low-power sources. Sources with dimensions of 1/4 and 1/2 of a panel have special terminals to feed the output voltage.



General view of the unified secondary power supply

SPS built by using high-frequency conversion and pulse regulation on the high voltage side (300-600V) with pulse regulators and voltage converters are promising for further development of the Unified System of Computers. In designing future power supply systems and units, consideration should be given to a 10-15 fold increase in the next five to seven years of the density of packaging of logic elements in third and fourth generation Unified System hardware built with matrix LSI circuits and LSI storage. This trend has to be considered in the design of SPS and power supply systems to ensure their electrical and technological design compatibility with the



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



electronics of the processing devices. In the process, advance development of the element base is becoming a topical question. The design of this base must be developed under a common plan coordinated among the countries that participate in the agreement on development of the Unified System of Computers.

Achieving specific indicators of 120-150 $\frac{W}{dm^3}$ required for the near future and increasing the efficiency factor to 0.75-0.8 primarily involve developing a highly efficient power element base:
transistors with working voltage to 1-1.5kV, switching currents to 10A, threshold frequencies of 10-20 MHz and turn-off time of 0.25-0.5 microsecond;
variable thyristors with turn-off time of 1-3 microseconds;
rectifier Schottky diodes with working currents of 30-100A;
electrolytic capacitors with high indicators for specific electrical energy ($5\text{-}50 \frac{FV^2}{dm^3}$ for compacitors with working voltage to 16V); and
integrated circuits for SPS control and protection circuits.

Using methods of high-frequency conversion and pulse regulation of electricity and development of a new element base will allow creation of highly efficient power supply systems for future computers in the Unified System based on SPS with puls regulators and high-frequency converters of main supply voltage with transformerless input.

COPYRIGHT: Izdatel'stvo "Finansy i statistika", 1981

8545
CSO: 8144/0138



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



DATA PREPARATION UNIT DEVELOPMENT IN BULGARIA

Moscow VYCHISLITEL'NAYA TEKHNIKA SOTSIALISTICHESKIKH STRAN in Russian No 9, 1981 (signed to press 21 May 81) pp 28-32

[Article by T. A. Topalov, candidate of engineering science, (People's Republic of Bulgaria)]

[Text] Bulgarian specialization in external magnetic media storage units allows Bulgarian specialists to solve a number of problems involving the design and production of data preparation units (UPD) [DPU] using magnetic media. In doing so, complex engineering problems have to be solved: development of electric motors with special characteristics, magnetic heads, contactless switching elements, methods of magnetic recording and others. Also, development of DPU's requires the optimal solution to a number of problems characterizing the "operator-device--medium-computer" system, for example, choosing the medium for data preparation and entry to a computer system.

The DPU's developed and manufactured in Bulgaria use standard magnetic tape with a width of 12.7 mm, recording density of 32 bits/mm and a floppy disk with a diameter of 203 mm. The key-to-tape units developed earlier than others have adequate capacity and fine quality and are produced in the socialist countries. They are compatible with computer external storage units. The floppy disk is compact and reliable. Key-to-disk units are characterized by random access and using them is promising for mini and microcomputer systems.

Used in developing DPU's is the experience of optimizing the data preparation process gained over many years of operation of key punch units, since the new DPU's must be sufficiently close in a systems sense to the key punch units. Otherwise, the new units would not be applied in the data processing process already established.

The first key-to-tape unit developed in Bulgaria was the YeS-9002. Compared to key punch units, work efficiency is raised considerably by the high speed of software and manual control of the data format input or varified from the keyboard, character display of current information, quiet operation of the unit and simple selection of the mode or function. The YeS-9002 gives the user the capability of achieving high economic indicators because of its high reliability, longer amortization time, time-saving computer entry operations, verification of data file preparation (which requires special devices with paper media), and the capability of reusing tapes. Produced over several years, the YeS-9002 is in successful use in a number of



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



countries. Modifications have been made to this unit, for example, the YeS-9002.02 can print out information from amgnetic tape.

The YeS-9004 is an updated version fo the YeS-9002; it has substantially improved functional, design and operating characteristics. It has a current data block display, based on a CRT with television scan or the "Elektronika VL100" and "Yunost' 603" consumer television receivers. This unit is very flexible with respect to the character set and recording density; for example, it is easy to modify the YeS-9004 for a recording density of 8 or 22 bits/mm. The user has the option of having the unit equipped with the functions for modulo 10 sum, modulo 11 sum and total sum of all fields.

The use in the YeS-9004 of circuits with an increased scale of integration, connectors with improved characteristics and a more successful design for the tape drive permits a substantial increase in the indicator for mean time between failures. The unit has a detached keyboard and smaller dimensions and weight. The capability of connecting a number of IO devices extends significantly its sphere of application.

The YeS-9004 has these modifications:

the YeS-9004.02, with output to the YeS-7187 printer;
the YeS-9004.03, with input from the IZOT 6001 card reader;
the YeS-9004.04, with a modem for data exchange over telephone lines;
the YeS-9004.05, with input from a perforated tape reader;
the YeS-9004.06, with output to perforated tape; and
the YeS-9004.07, with floppy disk storage.

The following table compares the YeS-9002, YeS-9004 and the MDS Data Record 6401 used in capitalist countries.

The devices operate in these modes: data entry, verification and retrieval; program entry and verification; and buffer storage input, verification and readout.

| Parameter | MDS 6401 | YeS-9002 | YeS-9004 |
|---|---|---|---|
| buffer storage | ferrite | integrated | integrated |
| element base | discrete | IC's | IC's |
| display | bit | character | block |
| tape drive | pressure roller | 1 roller | 1 roller |
| recording method | NRZ1 | NRZ1 | NRZ1 |
| density, bit/mm | 32 | 32 | 8, 22, 32 |
| temperature range, °C | 10-38 | 5-40 | 5-40 |
| weight, kg | 122 | 64 | 48 |
| dimensions, mm | 794x743x1092 | 582x641x583 | 481x396x483 |
| data validity, bits | -- | $10^7$ | $10^8$ |
| MTBF, hours | -- | 500 | 1000 |



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



In recent years, a complex of DPU's using floppy disks has been developed and is being assimilated in Bulgaria. Tests on the SM-6910 key-to-floppy disk unit have been successfully concluded. Prototype series of the YeS-9112.01 key-to-floppy disk unit and the YeS-9113 floppy-to-tape converter is being completed. All these units are built with microprocessor circuits as the element base. They are modular and have a common external shape.

The SM-6901 includes a rack with keyboard, electronic unit and power supply, cabinet with two floppy disk storage units and power supply, and a video monitor.

SM-6901 characteristics:

| | |
|---|---|
| element base | LSI circuits |
| ISO compatible | yes |
| programs | 10 |
| data validity | $10^7$ |
| MTBF, hours | 500 |
| weight, kg | 133 |

The SM-6901 performs the following functions: entry, processing, checking, retrieval by address, retrieval by content and end of data; initialization; count of total and check sum; copying of content of floppy, file, specific block; operation with chain of programs; statistics of operator work.

The YeS-9112.01 differs from the SM-6901 only by the codes of the characters in the DKOI [decimal interchange code]. Data prepared on the YeS-9112.01 and SM-6901 units can be entered into a computer directly from a YeS-5057 unit for Unified System computers and from external floppy disk storage units for the System of Small Computers.

The YeS-9113 converter can perform all the functions provided by the SM-6901 and YeS-9112.01 units and a number of additional ones:

finding a tape mark;
finding an end of file on tape;
retrieval by content on tape;
movement of tape forward;
movement of tape backward;
copy from floppy to tape;
copy from tape to floppy;
copy from floppy to tape on a specific condition;
copy from tape to floppy on a specific condition;
key-to-tape recording; and
deletion of data blocks on tape.

The YeS-9113 includes a rack with keyboard, electronic unit and power supply, cabinet with two YeS-5074 NGMD [floppy disk storage units], cabinet with SM-5300 NML [magnetic tape storage unit] and a video monitor.

The use of microprogram control in these units and external storage units with random access leads us to expect a further increase in operator productivity compared to that obtained with the YeS-9002 and YeS-9004 units.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



The YeS-9003 multiconsole key-to-tape system handles the complex problems involving data preparation and entry of files into a computers. Use of a general-purpose microprocessor in the system determines the implementation of a set of complex functions. The YeS-9003 system is built with the following hardware:

IZOT 310 CPU (32K 12-bit words storage);
SM-5400/01 NMD [magnetic disk storage unit] (6M byte capacity, 1500 rev/min);
SM-5300 NML [Magentic tape storage unit] (recording speed is 32 cm/s, density is 32 bits/mm, method is NRZ1);
IZOT 0232 operator console;
YeS-7186 mosaic serial printer (rate: 180 cps, 132 char/line);
YeS-0101 keyboard (16 keys); and
video monitor with display of 128 characters (4 lines of 32 characters each).

However, the high cost of the system units make it noncompetitive with standalone buffer storage units in configurations using less 8 to 10 consoles.

Studies have shown that operator productivity is increased from 40 to 120 percent when these units are used, compared to key punch operations, and data preparation expenses, which account for 30 to 50 percent of total computer operating cost, are substantially reduced.

Irrespective of the fact that on-line processing is more promising, data preparation in the batch mode will predominate in a number of cases. Naturally, for this it is necessary to continually raise the technical and economic indicators of the hardware employed. Efforts are underway in Bulgaria primarily in two directions: optimization of the operator-device interface by expanding functional, editing and processing capabilities and by improving operating characteristics, especially hardware reliability, by searching for new design and engineering solutions.

COPYRIGHT: Izdatel'stvo "Finansy i statistika", 1981

8545
CSO: 8144/0138



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



YES-1045 COMPUTER, CHARACTERISTIC FEATURES

Moscow VYCHISLITEL'NAYA TEKHNIKA SOTSIALISTICHESKIKH STRAN in Russian No 9, 1981 (signed to press 21 May 81) pp 32-41

[Article by A. T. Kuchukyan, T. Ye. Sarkisyan and I. B. Mkrtumyan, all candidates of engineering science (USSR)]

[Text] The YeS-1045 is a general-purpose computer with medium throughput designed to solve a broad range of scientific, technical, economic, informational and logic problems.

The logic structure of the YeS-1045 meets the requirements imposed on the logic structure of the second phase of the Unified System of Computers.

The YeS-1045 computer hardware is modular which allows a user to design specialized computer systems corresponding to the purpose of the system being created to the highest extent without making any changes to machine structure and design.

YeS-1045 computer hardware can be classified as standard, part of any system configuration, and optional, delivered upon customer request.

YeS-1045 standard facilities are:

facilities for processing in the basic and extended control modes;
the Unified System universal instruction set;
dynamic address translation [DAT];
indirect data addressing in channel;
instruction execution retry;
monitor facilities;
program event recording [PER];
fast 8K-byte buffer storage;
byte-multiplexer channel 0; and
block-multiplexer channels 1 and 2.

YeS-1045 optional facilities are:

floating-point instructions;
direct control facilities;
facilities for organizing a dual processor system;
block-multiplexer channels 3, 4 and 5;
a second byte-multiplexer channel (replaces block-multiplexer channel 4);



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



one or two channel-to-channel adapters;
logic repeater; and
configuration panel.

The basic set of the YeS-1045 computer is placed in three standard Unified System racks (processor and IO channels, main storage and power supply). The channel-to-channel adapters and logic repeater are placed in the same racks.

In the YeS-1045 computer, CPU control is hardware-microprogram with a machine step of 120 ns. YeS-1045 throughput using a Gibson mix for scientific and technical problems reaches 880,000 operations/second; for economic problems using a GPO mix it is 540,000 operations/second.

The CPU includes a microprogram control unit (BMU), instruction fetch and interruption servicing unit (BVK) and operations unit (BO).

The microprogram control unit is designed to control CPU and IO channel operation by means of microprograms stored in control storage (UP). Machine control storage is made with bipolar transistors and consists of two parts: permanent control storage (UPP) and loadable control storage (UPZ).

Permanent control storage (addresses with the ranges of 0-6 and 7-8K words) is designed for storage of CPU and IO channel control microprograms; loadable control storage (addresses 6-7K words) holds microprogarms for console operations and diagnostic microprograms. Loadable control storage can also be used to introduce special operations in solving specific user problems on the YeS-1045. Microprograms are loaded from the ML-45 magentic tape cassette storage console unit.

The instruction fetch and interruption servicing unit is designed to prefetch, store and partially decode instructions, fetch operands and service interruptions. The version adopted in the YeS-1045 computer processor is an instruction fetch unit with buffering which enables overlap of instruction fetching from the instruction buffer storage (BPK) or main storage with execution of instructions in the CPU.

Instruction buffer storage is made with integrated circuits and has a capacity of 32 8-byte words. It is designed to hold 64 words of instructions, prefetched from main storage, with the capability of their rapid fetching and moving into the CPU to fill three instruction buffer registers. These registers are filled after the contents of two of them are used. Instruction fetch time with this scheme is actually determined by the transfer from one buffer register to another. Because of this and the availability of the instruction buffer storage, instruction fetch efficiency is appreciably enhanced since practically the majority of problems make use of blocks of instructions with cyclic repetition within these blocks.

The operations unit consists of four- and one-byte data paths; it is designed to execute a broad set of arithmetic and logic operations used in the Unified System. A special high-speed processor-accelerator is provided in the operations unit to speed up execution of certain "long" arithmetic and logic operations. Executed in the accelerator are all multiply operations (except extended), binary-decimal translations, packing and unpacking, all primary shift operations, and some move and store operations, 25 operations in all. Thanks to incorporation of the accelerator in the operations unit structure, CPU throughput increased an average of 15-17 percent while equipment bulk increased only 5-6 percent.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



The main storage control unit (BUOP) organizes data exchange between the CPU, IO channels, array processor and main storage; provides for rapid access of frequently used data and instructions from the appropriate blocks of buffer storage without referencing main storage; converts virtual addresses into real; determines and signals violations of storage protection; and organizes a common extent of main storage of two computers in a dual processor system.

Buffer storage (BP) with a capacity of 8192 bytes is made with integrated circuits with data access time equal to a CPU step. The entire 16-byte block of data is kept in buffer storage when the CPU references main storage for some part of this block. Thanks to the algorithm used to fill buffer storage, the probability of finding the next operand in the buffer is about 75 percent, which reduces the average time for fetching data used most often by the CPU.

Virtual addresses are translated into real by using the special apparatus, the dynamic address translation facility (DPA) and the corresponding segment and page tables. This process is speeded up (intermediate references to main storage are excluded) by the translation-lookaside buffer (BBP), where the real addresses are written for the pages last referenced. Fetching a real address from the translation-lookaside buffer [TLB] does not lengthen the main storage reference operation. Provided in the YeS-1045 computer is the capability of simulataneous storage in the TLB of the results of translation for three blocks of virtual storage, which allows avoiding a complete purge of the TLB when a program is replaced. The DAT hardware facilities provide the capability of using up to 16M bytes of virtual storage.

The main storage adapter provides the interface between the main storage control unit and main storage, directly controls main storage operation in the read or write modes, and detects errors when data is read from main storage by using the modified Hamming code checking apparatus. Single errors are corrected by using the correction circuits; corrected data is sent from the adapter to the data exchange control unit. In the process, the data in main storage is not corrected.

Storage protection facilities provide protection of individual zones of main storage from unauthorized access. The protection mechanism provides protection in the store mode alone, and during both the store and read modes. A 2048-byte block is the protected unit in main storage. The main hardware protection facility used is the high-speed storage with a capacity to 4096 bytes, which provides protection to a maximum of 8M bytes of main storage.

There are two types of channels in the YeS-1045 computer: byte-multiplexer (BTMK) that can operate in the byte-multiplexer and burst modes; and block-multiplexer (BLMK) that can operate in the block-multiplexer and selector modes. The IO channels in the YeS-1045 are controlled by the combined hardware-microprogram method. Data is exchanged through the IO interface under control of channel hardware facilities concurrently with CPU operation. Data exchaneg between IO channels and main storage and processing of control information are performed by CPU facilities under microprogram control. IO channels share control storage apparatus with the CPU. During operation of IO channels under microprogram control, execution of the current instruction in the CPU is halted, and the MP [array processor] facilities and other CPU facilities are made available to the IO channel. When several IO channels request microprogram service as the same time, service is given based on the hardware priority scheme.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



Control words of the active subchannels for all six IO channels are stored in a section of local CPU storage with a 64-word capacity; each channel is allocated 8 words. The last 16K-byte area of main storage is used to store control words for passive subchannels. Up to 10 devices at a distance up to 60 m can be connected physically to each channel. A logic repeater is made part of the machine to extend the distance or number of devices connectable through the IO interface. Throughput of each byte-multiplexer channel is 47K bytes/sec in the multiplexer mode and up to 110K bytes/sec in the burst mode. Without considering data chaining, block-multiplexer channel throughput ranges from 0.5M to 1.5M bytes/sec as a function of the number of them. Total throughput of all IO channels is 5M bytes/sec.

The YeS-1045 computer has a highly develoepd checking and diagnostic system. The main components of this complex of facilities are hardware-microprogram recovery of 179 instructions when random failures occur, microdiagnostics that enable checking all computer facilities in the static and dynamic modes, central checking and measurement of power supply, and automatic preventive checking and diagnostics of faulty integrated circuits in TEZ's [standard exchange cards].

Yes-1045 computer recovery facilities enable continuation of computer operation when single errors occur in main storage, bypassing the consequences of failures in the CPU and channels through hardware-microprogram retry of operations, and recovery f the operating system with the aid of software facilities.

The YeS-1045 computer can also operate with a reduced amount of buffer storage when a part of it fails.

When the check circuits detect a machine error, control synchropulses are blocked, i.e. execution of instructions at all levels of overlap is terminated, and hardware logging of computer status is performed in the control console storage (PPU) with a capacity of 512 bytes. Hardware logging in the control console storage is performed during the first and eighth retry attempts. Control console storage allows preserving the computer status at the time of the malfunction when servicing of interruptions from the check circuits is suspended, and the sufficiently large capacity of it provides the capability of having information on the status of practically all machine flipflops and registers. After hardware logging is completed, CPU and channel registers are purged, the possibility of retry of the interrupted operation is analyzed. The retry method depends on the type of operation, computer status at the time fo the error and the type of error.

Checking for change in initial data is provided for in microprograms of instructions. When the data is changed, the microprogram branches to alternate execution of the instruction. Preservation of initial data is provided for to recover computations during a malfunction. The following facilities are used for this in the YeS-1045:

blocking of change of the first operand in RR and RX format instructions upon an error signal, which permits dispensing with storage of initial data;
storage of address of second operand in work area of local storage and recovery of it during retry, which permits dispensing with storage of base register contents;
storage of the changed part of the first operand with subsequent recovery of it during failure in decimal arithmetic instructions;
storage of the count of successfully processed bytes in RR format logic instructions with continuation of processing upon failure with retry of processing of one byte;



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



checkpoints in microprograms of certain instructions, thanks to which upon retry, the successfully executed part of the instruction is not repeated.

In the YeS-1045 computer, hardware-microprogram recovery of IO instructions is possible to a certain threshold, determined by the start of communication with a device through the interface. If the threshold of hardware-microprogram retry of an IO instruction is passed, an IO interruption is executed and the usual and expanded status word containing the data specifying the location and condition of the error is stored. After this, recovery is performed at the program level by using the CCH and ERP programs. The CCH program at the same time generates a record in the log file.

In the YeS-1045 computer channels, retry of channel commands is also provided for at the request of a device equipped with the special facilities for issuing this request to a channel. Thanks to this, when an error occurs in a device during execution of a chain of commands, the channel supports reissue of the last command upon request from the device.

Hardware diagnostic facilities include the service adapter and the ML-45 console storage unit. The service adapter enables finding and reading of data from the console storage unit, accumulation of data, decoding of diagnostic operations, execution of microdiagnostics and comparison of computer status with a reference standard. To simplify the loading path, data from the console storage unit goes to the service adapter in series code and after its translation into parallel goes to the CPU circuits. This small amount of hardware facilities for the diagnostic unit (seven TEZ's [standard exchange cards] for the service adapter and five for the electronic part of the ML-45) substantially increases the operational reliability and repairability of the YeS-1045 computer. To check the hardware part of the computer, the service adapter makes use of data paths and microoperations that exist in the computer.

Yes-1045 computer diagnostics consist of three stages.

At the first stage of the microdiagnostics, control of the halted CPU is monopolized by the service adapter. The CPU executes individual microoperations upon orders from the service adapter. To poll computer status, the service adapter uses data serialized with the aid of a multiplexer, i.e. data on the status of computer registers and flipflops in the form of a serial stream of bits. Each data byte in the serial stream has its own address and during the microdiagnostic procedure is fetched from the general stream to the data register in the control console; signals output from here go to the circuit for comparison with the reference value kept in the diagnostic register in the service adapter.

The standard procedure in first-stage microdiagnostics includes:

loading of a microinstruction from the console storage unit into the data register in control storage and execution of it. Test sets are defined by the specifying field of the microinstruction. Executed in this mode are the microinstructions needed to check the functions of the assembly being tested;
reading from the PNK [expansion unknown] of the address of the register (in the general stream of serialized data, the content of which is compared with the standard, and the reference standard itself;



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



comparison of the result with the reference standard in the comparison circuit SERAD [expansion unknown];
halts with display of the halt number on the control console lights. The halt number is used to locate the comments in the microdiagnostic printouts that contain the identification of the improperly functioning flipflop, list of faulty TEZ's [standard exchange cards] and recommendations for locating the malfunction.

In the error halt status, service personnel can obtain additional information on the system indication.

Comparison of the low-order bits of the address of control storage with the reference is provided for to check the microoperations of analyses in the diagnostic apparatus. This permits identifying the microoperations of analyses of modification of control storage address bits that enable branching according to the various computer states. First-stage microdiagnostics also check the facilities for manual control fo the YeS-1045 console by simulating actions of keys and switches. Convenient facilities for test cycling and microinstructions in the microdiagnostic test are provided to facilitate finding malfunctions in the diagnostic unit.

Checked in the first stage of microdiagnostics are all hardware facilities of the CPU, channels, storage control, some fo the storage units, console typewriter ar channel-to-channel adapter. Apparatus is checked in microoperations at the rate specified by the rate of reading of data from the console storage unit, which simulates the "rapid" step mode more closely than the actual machine rate of executing microoperations. Therefore, tests in the first-stage microdiagnostics cannot identify some errors associated with the timing diagram of operation of computer units.

Second-stage microdiagnostics, making use of loadable control storage checked in the first stage, identify errors associated with the timing diagram. In this stage, microdiagnostics are performed under control of the computer itself at the actual machine rate the following way:

a diagnostic microprogram is input to control loadable storage from the console storage unit and receives control after loading;
the diagnostic microprogram checks computer assemblies in the diagnostic mode, i.e. at the actual machine rate; after storage of the check results, control is returned to the service adapter for analysis of the check results by comapring them to the reference standard under control of the service adapter.

The set of microdiagnostic etsts is fixed and designed for the maximum configuration. The type of channel connected under a specific number or its availability in the system is determined by microdiagnostics by polling the special facilities of the channel that store this information. When some channel is not used in a system, the microdiagnostic tests for this channel are automatically skipped. The amount of main storage in its microdiagnostics is determined by the condition of the invalid address that enables checking the amount of storage that is available in the system. Increasing the amount of storage with such organization of the microdiagnostics does not require changing it.

The first two stages of microdiagnostics permit checking and localizing a malfunction in the YeS-1045 computer hardware facilities, including all storage devices.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



The third stage of computer diagnostics is performed by using the facilities of functional (software) diagnostics. The purpose of this stage is to check and detect improperly functioning system facilities: the system of instructions for external devices, external device control units and concurrent operation of devices. The YeS-1045 computer functional diagnostic facilities are teh basic test, the set of test-sections executed under control of the diagnostic monitor, and the set of autonomous diagnsotic tests. Checking the functional facilities of the system begins with a check of the nucleus of the instruction system by using the basic test. Hardware facilities in the diagnostic unit permit loading the basic test into main storage from the console storage unit. The basic test checks the group of CPU instructions, IO instructions and the paths for communication with the standard external devices-tape drives and the typewriter. Further checking of system functional facilities is performed in the test-sections of the KPTO [maintenance program complex].

Computer operating experience indicates that localization of failures in the power supply system and fans, places of excess heat, and preventive checking of the voltage of secondary power supplies, the number of which is several dozens, require considerable time. To obtain the required parameters of operating reliability in the YeS-1045 computer, there is a system for checking and diagnosing power supplies which performs the following functions:

automatic checking and measurement of output voltages from power supplies;
automatic and manual setting of preventive levels of power supply voltages (UVIP);
localization of the place and type of emergency malfunctions, such as hot spots, fan failures and power control unit (BUP) failure with audible and light warning signals.

In measuring voltages output from UVIP [unified secondary power supplies = USPS], the measuring switch connects the outlet of the selected USPS to an analog-to-digital converter of a digital voltmeter; information from this goes to a comparison circuit.

When USPS voltage deviates beyond tolerances, a signal is generated that goes to the display.

Preventive five-per-cent levels of USPS voltages are set either automatically through the 12-channel interface from the diagnostic unit or maunually from the system console. The USPS address goes to the decoder; after decoding it controls the switches for the preventive check, to the input of which also comes the condition of the level of $\pm 5\%$.

An audible alarm signal is generated and the address of the malfunction is displayed when overheating occurs, fans stop or the power control unit malfunctions.

A basic problem in operating computers with incomplete ZIP [kit of spare parts, tools and accessories] is repair of faulty TEZ [standard exchange cards = SEC]. Manufacturers have to set aside up to five percent of their capital, investing it in ZIP's, and the cost of storing them in warehouses is rather high. Also, the volume of repeated return of faulty SEC's has increased. It is estimated that about 50 percent of the SEC's is returned to the user without repair since no malfunction is detected in them. It is cheaper to repair the SEC's on site. This is possible with the YeS-1045 computer because of the availability of the integrated



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



autotester, diagnostic tests, references that are part of the operating documentation and the PET-1 tester. The YeS-1045 computer autotester uses the facilities of diagnostics and the console storage unit to test SEC's. The advantages of the autotester are its low cost (it has only three SEC's), simplicity and convenience of maintenance, detailed diagnostic documentation, quick recovery, low demands on the classification of maintenance personnel and the capability of reducing ZIP's.

Trial operation of the YeS-1045 has indicated that the checking and diagnostic facilities described above substantially raise computer operating efficiency and may be recommended for application in other machines.

The YeS-1045 computer structure allows setting up dual processor and multimachine systems. Dual processor systems are established on the basis of a common extent of main (up to 8M bytes) and external storage for both processors and peripherals operating under the control of a single operating system. A special configuration panel combined with the appropriate equipment provides the capability of controlling the modes and configuration of the dual processor system.

Multimachine systems are set up by complexing at the channel level by using the channel-to-channel adapter, direct control facilities, a common extent of external storage and common peripherals; each computer operates under control of its own operating system.

The array processor (MP) is an option for the YeS-1045. It is a specialized computer connected to the YeS-1045 computer as the seventh channel. It is used to process numeric series encountered in problems where repeated operations have to be performed on a group of data (series, arrays), thereby relieving teh load on the CPU.

The array processor is organized for parallel processing of data in the CPU and array processor. It has three units: control, arithmetic and microprogram control. As a conveyor, the array processor executes arithmetic operations according to the expression $z = y \pm vx$. Vector operations, spectra correlations, complex multiplication and fast Fourier transforms are performed in the array processor. Its basic cycle is 155 ns and it is fitted in a standard Unified System rack.

The array processor is used by a set of access method programs that expand the capabilities of YeS OS.

The high throughput of the computer complex made up of the YeS-1045 and the array processor permits a substantial increase in the efficiency of solving problems on pattern recognition, processing of geophysical data, etc.

COPYRIGHT: Izdatel'stvo "Finansy i statistika", 1981

8545
CSO: 8144/0138



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



VIDEOTON FAMILY OF INTELLIGENT ALPHANUMERIC TERMINALS

Moscow VYCHISLITEL'NAYA TEKHNIKA SOTSIALISTICHESKIKH STRAN in Russian No 9, 1981 (signed to press 21 May 81) pp 41-47

[Article by L. Nylas, G. Hudoba and J. Bornemissza, all engineers from the Hungarian People's Republic]

[Text] Products from the Videoton computer hardware plant, which has a decade of experience in manufacturing computer hardware, are in great demand among both Hungarian and foreign specialists. The popularity and wide distribution of Videoton devices is due not only to their technical parameters, but also to high quality and fine esthetic and human engineering characteristics. An example of the results achieved by Videoton is the family of intelligent alphanumeric terminals based on microprocessors.

Microprocessors were developed to simplify data processing and control hardware, increase its reliability and reduce cost while raising functional flexibility at the same time. Microprocessor throughput approaches that of early small computers. However, to use them as computers requires expanding main storage, connecting external storage units, providing facilities for interruption processing, etc.

Microprocessor system software development is proceeding in two directions: first, microprocessor software is being developed on so-called HOST machines, and auxiliary software for development, such as Assembler and Simulator, is run on this machine; second, microprocessors are being made capable of software devlopment. Videoton chose the latter path.

In 1974, microprocessors began to be used at the plant in data transmission and processing systems. The VTS 56100 unit with microprocessor control dependent on a built-in program was developed at that time and used as a synchronous or asynchronous terminal. From a functional point of view, the most advanced model in the VTS 56100 terminal line is the office video terminal with two external storage cassettes. It can be programmed by the user, used in large systems, provides extensive offline functions for data base maintenance, storing and editing; because of this, this unit is recommended primarily for use in the nonindustrial sphere.

In 1976, development began on two, essentially new display families which then went into series production. The development was based on the experience gained in the development, production and use of the well-known VT 340 display (YeS-7168 and SM-7206), as well as the VTS 56100 microprocessor family of video terminals, and



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



with regard to the requirements and capabilities specified by the Unified System and System of Small Computers programs.

The one family is built with basic devices with small power consumption, made with highly reliable LSI circuits and they provide great functional flexibility and broad capabilities. The early devices in this family are the SM-7219 terminals (which passed international tests within the framework of the System of Small Computers in 1977), the YeS-7168M terminals and improved YeS-7168 terminals.

Devices in the other family are terminal stations: intelligent alphanumeric video terminals with a set of external devices, easily programmable by the user in Assembler or a high-level language. Two devices in this family have been approved at the international level: the SM-7401 which has a built-in minicassette storage unit (which passed joint tests within the framework of the System of Small Computers in 1978), and the VT 20 equipped with large disk storage. These devices meet world standards in reliability and functional capabilities in their category.

A measure of the intelligence of these programmable termianls based on microprocessors is the aggregate of these capabilities:

expanded random access storage that enables loading and running of various user programs;
connection of high-capacity external storage units, forexample, NML [tape drives], NGMD [floppy disk units] and NMD [disk] with permanent or replaceable disks that support processing of data files;
connection of paper-tape IO devices to maintain compatibility with older systems;
connection of serial or parallel printers; and
connection of various parallel and serial data transmission channels.

The capability of connecting external devices to an intelligent alphanumeric terminal is shown in the block diagram (fig. 1). Naturally, all connection capabilities cannot be used at the same time in one device; besides arrangement problems, this would cause a shortage in the time resource for the central unit. However, through suitable selection, one can build a configuration most suited to a specific application.

Providing the circuit capabilities listed above and others still does not make a machine intelligent: Without an efficient software system, all this would remain just a possibility.

The basic unit in a Videoton intelligent alphanumeric terminal is a microprocessor minicomputer structurally integrated with an image generation system. Let us list the components of the block diagram of the device (see fig. 1) and their characteristics.

Central processor:
- byte processing of data
- instruction execution time of 2-9 microseconds
- seven internal registers
- interruption processing
- three internal timers
- storage, expandable to 64K bytes.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



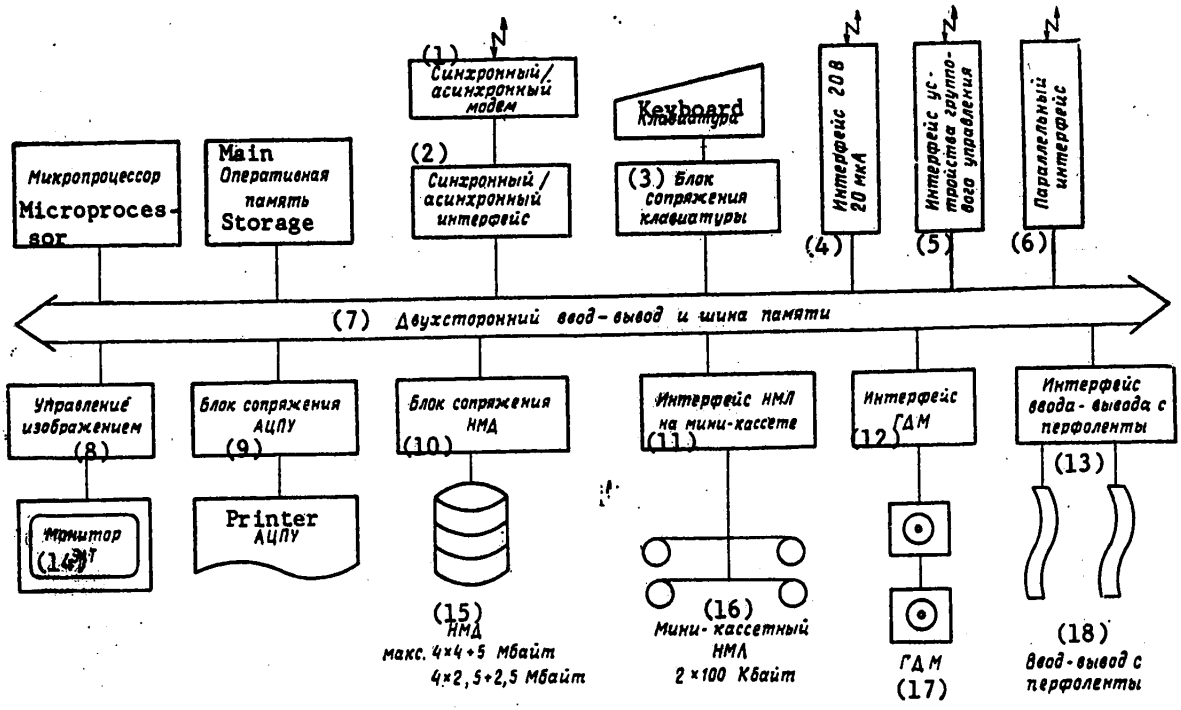



Fig. 1. Consolidated block diagram of family of intelligent aphanumeric video terminals

Key:

1. synchronous/asynchronous modem
2. synchronous/asynchronous interface
3. keyboard interface unit
4. 20V 20 microamp interface
5. cluster control unit interface
6. parallel interface
7. two-way IO and storage bus
8. image control
9. printer interface
10. disk interface
11. minicassette tape interface
12. GDM [floppy disk] interface
13. papertape IO interface
14. CRT monitor
15. disk: maximum of 4 x 4 + 5M bytes, 4 x 2.5 + 2.5M bytes
16. minicassette tape storage, 2 x 100K bytes
17. floppy disks
18. papertape IO

Display:

10 x 12 dot matrix
2 x 64 alphanumeric characters
64 special symbols, for example drawing of lines
Greek letters etc.
screen capacity 25 x 80 or 24 x 80 characters
field and line image organization.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



| | |
|---|---|
| Keyboard: | standard typewriter style arrangement<br>separate numeric keypad for data processing problems<br>function keys<br>upper and lower case<br>automatic repeat of characters<br>audible signals. |
| Data transmission line: | synchronous or asynchronous<br>interface according to recommendation 24 of the MKKTT [International Telephone and Telegraph Consultative Committee] or a current loop of 20V 20mA (S2, IRPS)<br>cluster control unit on asynchronous line<br>maximum rate withs erial data transmission is 9600 baud with the capability of programming<br>parallel interface<br>programmable process for data transmission and error protection. |
| IO devices: | parallel or serial matrix printer<br>card reader<br>card puncher |
| External magnetic storage units: | two minicassettes with capacity of 2 x 100K bytes<br>two floppy disks with capacity of 2 x 250K bytes<br>replaceable disks with capacity of 5 + 5M bytes or 2.5 + 2.5M bytes. |

External magnetic storage units can be optionally connected to the basic device.

Operating System. The intelligent alphanumeric video terminal (PAV) and the VT 20 unit can be used first of all as standalone units easily programmable by the user. The difference between them is that the PAV has two built-in minicassette tape storage units, while the VT 20 operates with 10M bytes of disk storage. Therefore, the operating systems for these devices differ somewhat because the VT 20 must make efficient use of the advantages and capabilities offerred by disks. The general scheme of the software system (SPO) is shown in fig. 2. A more detailed general plan of the system is shown in fig. 3.

The basic software for the programmable devices has the following programs:

resident program. The programmable read-only memory (PPZU) [PROM] contains the permanent program that assumes control automatically when the device is switched on. It performs the initial adjustment and provides the capability of loading and starting programs written on magnetic tape or disk;
monitors. The device has three monitor programs. The first debugs programs and corrects errors; the second has functions of programming in the PROM; and the third can control test programs. The programs permit loading and startup of programs of various formats, changing storage contents, printing storage contents, output of storage contents on data medium in loadable format, and funtions of programming the PROM;
programming languages. The Assembler translator translates programs written in Assembler and indicates syntactic errors. It works in two passes, providing thereby use of postdefinition labels;



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



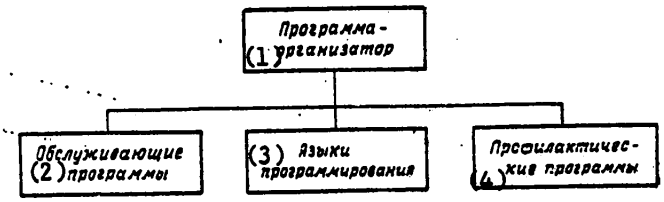



Fig. 2. General scheme of software system

Key:
1. organizer program
2. service programs
3. programming languages
4. preventive programs

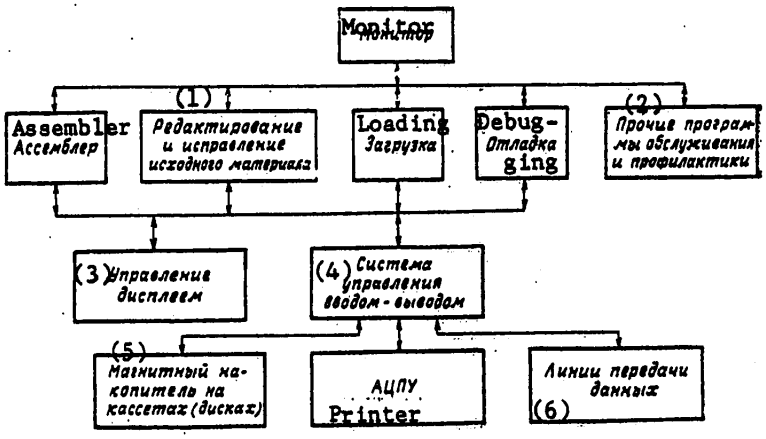



Fig. 3. General plan of system

Key:
1. editing and correction of source material
2. other service and preventive programs
3. display control
4. IO control system
5. cassette (disk) storage
6. data transmission line

Basic interpreter provides the capability of running programs written in Basic. The interpreter uses optional circuit capabilities of the device. Also, the instruction set is expanded by operations on operations on character strings; editing and correction of source material is used to create alphanumeric files that are recorded on magnetic tape or disk by means of the keyboard or display screen and to change source alphanumeric files;
IO Control. The VT 20 operating system with time sharing enables execution of



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



of logical control of external devices;
preventive programs. These include the test programs that check all device units and the test monitor that controls them;
other programs. The device has numerous programs used to solve problems often encountered in practice. The most important of them is the data file management system on disk that was developed for office systems.

The efficient operating system, numerous user programs and flexible schematic design, the intelligence of the device, enable use of these devices in all sectors of economics, starting with devices for data transmission and processing to desktop computers.

Recommended Areas of Application. In query applications, the device furnishes the operator rapid access to data bank files kept on disk. The device can be used to update files and for inquiries in teh interactive mode.

In data writing applications, the video terminal replaces traditional units, for example a card puncher, and can be used to update source files and store data in the online or offline mode, making use of the advantages offered by the display and external magentic storage units.

COPYRIGHT: Izdatel'stvo "Finansy i statistika", 1981

8545
CSO: 8144/0138



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



YES-2335 ARRAY PROCESSOR FOR YES-1035 COMPUTER SYSTEM

Moscow VYCHISLITEL'NAYA TEKHNIKA SOTSIALISTICHESKIKH STRAN in Russian No 9, 1981 (signed to press 21 May 81) pp 47-50

[Article by G. P. Nikolov and V. D. Lazarov, candidates of engineering science, and P. P. Daskalov, Y. V. Ivanova and K. D. Kirov, scientific associates, all from the People's Republic of Bulgaria]

[Text] Constantly growing is the number of users of computer hardware for performing rapid and efficient digital processing of large files of data represented in the form of arrays or vectors. Processing of such data files is required in seismic exploration (geophysics, searching for oil, gas and mineral resources); in studies of turbulence (meteorology, oceanology, aerodynamics); in radiolocation and processing of results of photography for enhancement and processing of images; in space exploration (telemetry, trajectory control); in research in nuclear physics; in matrix arithmetic; in mathematical statistics; for solving differential equations and digital processing of signals.

However, modern general-purpose computers, despite the continuous increase in speed, are not capable of meeting the requirements of throughput and system computing capacity for these types of problems. Execution of repeated arithmetic operations on elements of large files of data is practically unrealizable. Thus, it is necessary to develop specialized computer facilities to solve these problems.

There are several ways of developing efficient array computers. One of them, in our opinion the most successful at the current stage of development fo the Unified System of Computers, has been implemented in the YeS-2335 array processor (MP). It was shown in 1979 at the exposition entitled "Facilities of the Unified System of Computers and the Small System of Computers and their Application." Let us consider some characteristics of this processor.

The YeS-2335 model-dependent specialized peripheral processor performs high-speed calculations within the YeS-1035 computer system. Processing in the array processor is parallel and independent of that in the YeS-2635 CPU.

The YeS-2335 connects to the YeS-2635 CPU through a specialized adapter that replaces a selector channel in the CPU. The adapter and array processor are linked by a specialized IO interface.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



From the system aspect, the array processor and adapter can be considered a complex IO unit combining the functions of a channel, control unit and external device. This complex is controlled by using the same system of control words used in the Unified System IO system (IO instructions, KSK [expansion unknown], USU [unit control word], SSK [channel status word]) and the IO interruption mechanism.

Data processed in the array processor are represented in one of the following formats used in the Unified System of Computers: fixed point, direct code, short format; fixed point, two's complement code, short format; floating point, short format.

Array processor operations can be divided into the following groups:

a) vector move operations: data are moved from one area of main storage to another with execution, if indicated, of format translation (from fixed to floating point, or vice versa);

b) matrix arithmetic operations: scalar multiplication, element-by-element vector multiplication, element-by-element sum of vectors, sum of vector elements, partial matrix multiplication, sum of squares of elements and square of matrix with sign;

c) matrix scanning (to determine value and position of minimum or maximum element);

d) complex multiplication;

e) solving difference equations;

f) signal processing operations: convolution of addition, convolution of multiplication, fast Fourier transform;

g) quadratic interpolation.

The array processor has the following assemblies and units:

unit for communication and fetching of operands; it effects communication with the adapter and YeS-2635 CPU, processes addresses of operands and control words and fetches operands;

arithmetic unit; it executes conevyor arithmetic processing of floating point numbers;

buffer storage; it holds operands, intermediate and final results and performs the functions of a buffer between the inetrface and arithmetic unit;

control storage; it holds microprograms for the operation of the communication and arithmetic units;

microprogram control unit.

The basic parameters of the YeS-2335 are:

| | |
|---|---|
| machine cycle | 200 ns |
| throughput | 5 million operations/sec (multiplication and addition) |
| throughput of interface between adapter and array processor | 3.5M bytes/sec |
| throughput of arithmetic unit | 20M bytes/sec |
| buffer size | 2 x 32 words |
| control storage size | 2K words (of 96 bits each) |



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



| | |
|---|---|
| maximum operand size | 64,000 elements |
| element base | TT-1.2 and 3 |
| structure | one cabinet with two frames |
| power consumption | 1 kW |

The basic software for the YeS-2335 array processor consists of methods of access (MDMP) and the resident module. It is implemented in the Assembler language with use of the capabilities of the macrolanguage and conditional assembly statements. The access method is activated by a user program in the FORTRAN, Assembler or PL/1 languages by using the CALL statement having the specified format. The resident module enables processing of SVC interruptions with a code specifically for the array processor.

The access method has the following parts: control phase, syntactic analyzer, channel program designer, dispatcher for queue of orders [zakaz], issuing of messages and processing of interruptions.

In addition to the basic, the array processor softwa e also includes a signal processing program system. It is a package of subroutines that are used in the analysis of various types of signals in digital form. The package contains subroutines that syntheisize filters by specified criteria for their characteristics, and subroutines that help evaluate the efficiency of the filters.

The lack of sufficient experience and the accepted technique of evaluating throughput of systems with the use of array processors prevent determining the efficiency of their application and fixing the class of problems in which this application is most expedient. To derive approximate estimates of the increase in system throughput, experiments were conducted in which the same specially selected examples of data processing were executed on a YeS-1035 computer with and without use of the YeS-2335 array processor. They confirmed the preliminary estimates that by using the array processor, throughput is raised by one-two orders.

Results of one experiment are as follows. An average-size seismic route was divided into 2000 quanta. The array processor was used for certain basic processings such as deconvolution before and after summing, filtration before and after summing, and use of a fan-shaped filter for 10 percent of the data. In this case, processing time by the CPU decreased by 65.6 s for the route. Annual processing of 300,000 24-channel recordings by using the array processor takes about 4,000 hours. Consequently, 135,000 hours are required for a system without the array processor, i.e. throughput is increased 34-fold.

COPYRIGHT: Izdatel'stvo "Finansy i statistika", 1981

8545
CSO: 8144/0138



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



SYSTEM FOR DESIGN OF APPLICATION PROGRAM PACKAGES FOR CALENDAR-SUBJECT PLANNING AND MANAGEMENT FOR UNIFIED SYSTEM COMPUTERS

Moscow VYCHISLITEL'NAYA TEKHNIKA SOTSIALISTICHESKIKH STRAN in Russian No 9, 1981 (signed to press 21 May 81) pp 51-57

[Article by V. A. Matulis, candidate of physical and mathematical science, and A. A. Chaplinskas, engineer, both from the USSR]

[Text] The "Vil'nyus" system for design of application program packages (PPP) f calendar-subject planning and management of development work is based on systems of network models for a multilevel automated system for planning and management of a machine-building industry [1]. The problems solved within this automated system form a unified complex of problems; therefore, they must be correlated with each other. Correlation is provided by the informational, organizational-methodological and program-mathematical compatibility of the problems. Also provided for is the use of uniform indicators, a unified system of classification and coding of technical and economic information and unification of reports output. Taken into account are the presence of information exchange between the individual complexes of problems and the intercorrelation of problems being developed (structure of plans, indicators, objects and processes of planning, resources, organizational structure and others). The system is based on principles close to those used in program generators for process control [2].

The "Vil'nyus" system is parametric [3]; its nucleus unites procedures that implement input languages and data base management for the given class. In constructing PPP, in the system input language, the user describes the package input language, the data base structure and other parameters.

The basic concepts of the "Vil'nyus" system and its implementation methods are discussed in more detail in [4, 5].

Requirements for Packages in the Given Class. Studies have shown that PPP for calendar-subject planning must include:
a nonprocedural input language that permits formulating the aims of the design in the form of a specification for deriving a certain set of output documents;
facilities for automatic planning and implementation of the plan for solving problems;
facilities for automatic adjustment of the plan for solving problems as a function of results of intermediate calculations;
facilities for input of initial data that provide the capability of punching data



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



directly from user working documents;
data base indicators that permit determining the current status of each of the data segments; and
facilities for building a temporary working model of the planned object that allows keeping in the computer undeleted complete information on the plan throughout the entire plan period and permits making test calculations to determine the consequences of some plan adjustment before its official approval or multivariant calculations in drafting a plan for a new period.

The PPP must provide the capability:
of several related PPP using a common data base; problem modules must be independent of the data base structure;
of segmenting data in a data base (by subjects, subdivisions, etc.) and arranging datya within a segment in the form of a heirarchical sequence of records;
of specifying a model of the subject in the form of a network that has several initial and objective events and permits directive restrictions on the time periods for completion of any specified events;
of including in the model linear operations or specifying them in the form of a linear graph; and
of automatic adjustment to the computer configuration and parameters that reflect the specific nature of specific user problems.

In addition to what is listed above, there are a number of other aspects that affect mainly the content of the library of problem modules and its functional capabilities. They are not considered in this work since they are not essential in defining the basic concepts of the system part of a PPP.

System Composition and Structure. The system includes the monitor, the set of problem-oriented subroutines and macro instructions, data base management programs, library of system tables, library of problem modules and the language for describing the structure of the PPP for generation facilities.

Internal control of tasks and jobs is performed by the monitor.

The monitor, one of the main components of the nucleus, performs decoding of tasks, forms the plan for solving a problem, checks the authority of the user, implements the plan for solving a problem, checks the informational provisioning of the computing process, adjusts the plan for solving a problem and manages the listing of its solution.

The "Vil'nyus" system has a modular structure of three levels. Subroutines and macro instructions make up the lowest level of the problem modules. In essence, they form a certain macro language used to program modules in the next level, the macro blocks. Macro blocks are problem modules defined in terms of algorithms for solving problems in the subject field in question. Only modules of the highest level can be described in terms of the subject field itself; these are processors that are organized sets of macro blocks.

The scheme of the potentially permitted calculations of any PPP for the class in question is represented in the form of a sequence, ordered with respect to the time axis, of segments that are described by oriented acyclic graphs with one terminal and one initial vertex. Some processor is set to correspond to each segment, and



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



some macro block is compared with each of the vertices (except the initial and terminal) of the oriented graph. The arcs define the potentially permissible order of execution of the macro blocks. The processors are described by tables that specify the topology of the oriented graph, establish the method of mapping the set of its vertices in the problem module library and determine the input and output processor data flow.

The data base management programs provide for loading and updating of a data base, creating a working model for test calculations, internal translation of data and output of results [6].

Problem modules do not have direct access to a data base. They operate with working sets of processor data formed by the facilities for internal translation of the data prior to execution of the process. After its execution, the results of the calculations are moved from the work sets to the data base by using the same facilities. Formation of work sets and recording of results are effected upon monitor requests issued in accordance with the plan for solving the problem and the description of the processor data flow. The description of the data base structure and work set structure is kept in the form of special system tables, which are used also in printing out the results, considered as one of the types of internal translation of data. Tables describing patterns of punching the source documents are optionally used in data input and update.

The problem of loading and updating a data base is considerably more complicated than that of internal translation of data. This is due in particular to the fact that the correspondence between source documents and data structures being generated cannot be established by using apriori descriptions, but must be defined by software. Input and update programs are organized as an independent processor included in any package being designed. The tasks for this rpocessor also include file management, operation with unconfirmed updates and formation of a working model used to conduct test calculations.

Operation of the monitor and data base management progarms is controlled by system tables; the main ones are the catalog of macro blocks, catalog of processors, requisite description table, file description table, punch pattern description table, table of messages and input language description table.

Several interacting PPP may be built on the same copy of the nucleus. Each of them may have its own input language, scheme of potentially permissible calculations and data structures and conduct a protocol for solving a problem in its own terminology. Consequently, each of the packages has its own corresponding set of system tables. All sets are organized into a common library. The packages operate by using the common library of problem modules. However, each package has its own procedure for starting the monitor and operates under control of the set of system tables determined by this procedure.

Formation of system tables is one of the main stages in designing PPP by using the "Vil'nyus" system. This stage is completely automated. A special language has been created for this purpose [4, 6] that permits the developer to describe in a convenient way the package input language, data base, patterns for punching the source documents, processors, texts of messages to be issued and the calendar (number of work days in a week, method of determining work Saturdays, units of



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



measurement of the time interval, etc.). The system tables are generated according to these descriptions by using software facilities included in the nucleus. To describe the package structure, one has to know well the principles for designing the structure and functioning of a PPP, upon which the "Vil'nyus" system is based. No knowledge of the internal design of the system itself is required.

The generation facilities are intended for adjusting a system to a specific computer configuration and specific nature of user problems. For this purpose, the apparatus of cataloged procedures and facilities of the macro assembler are used. Computer parameters and problem parameters are described by using the generation form. A special program, operating directly under control of YeS OS, examines the procedures and macro instructions, adjusts them according to the values of the parameters specified in the form, performs translation of the system and creates the data sets needed. A working version of the system is generated as a result.

Problem Modules. To cover the functions of the problem area in question, in the authors' view, the library of problem modules must contain modules for calculation of time characteristics of the network and load of the executives and modules for balancing the plan by time, cost and other indicators, distribution of resources, formation of plan indicators and expeditious control.

Development of modules of these classes for the machine building sector is a complex and laborious task. Standard modules only for calculation of time characteristics have been included in the problem module library of the "Vil'nyus" system. They are oriented to models of subjects with a capacity up to 16,000 operations; maximum number of subjects is 999. Linear operations are also allowed in the models along with network. A network may have an arbitrary number of initial and objective events. Directive restrictions are permitted on the term for completion of any events in the network. The type of networks is "operation-arc."

The set of problem modules permits:
combining individual fragments (subnetworks) into a common network;
performing analysis of a network as a whole or by fragements to detect structural errors and circuits;
calculating time characteristics by using a mixed linear-network model;
removing executed operations and automatically determining terms of initial events of an adjusted network according to report information;
isolating from a network a subnetwork that enables completion of some set of specified (check) events and calculating the time characteristics of this subnetwork;
forming according to a specified set of indicators in a specified profile output documents on time characteristics of operations, time characteristics of events and executed operations.

Dates of initial or objective events in the network may be used as the source in calculating time characteristics. Machine-oriented algorithms given in [7] were used in designing the modules.

Designing Packages. To design a package by using the "Vil'nyus" system, it is necessary to generate a system that takes into account the computer configuration and the specified values of the problem parameters of the problems to be solved, to write a procedure to start the monitor, to describe the package structure in the "Vil'nyus" system language, and to expand the library of problem modules with the required macro blocks.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



In the case of expanding the subject area with new concepts, it is necessary to write additional subroutines for the meaning translation of data. Subroutines are called during entry of data which describe the new concepts. To include the subroutines in the package, they must be described in the system input language.

System Use Experience. Let us cite two examples of using the system as an illustration of system capabilities.

Example 1. A program package to generate on a computer a five-year subject plan for development work in the sector. Thsi package is designed to formulate five-year plans for development by the sector of technical complexes under the conditions of the restriction on resources available to the organizations and enterprises and the amounts of state budget appropriations allocated to the sector as a whole for development work. Taken into account in the process are the relative importance of the development work being undertaken and the minimum permissible and maximum possible rates of development of the individual technical complexes, their parts and systems.

Included in the complex of problems for five-year subject planning of development work in the sector are:
calculation of technical and economic indicators and generation of the initial alternative for the five-year subject plan for sector development work;
optimization of the volume indicators and generation of an optimized five-year plan for sector development work;
calculation of the development effort workload on organizations and enterprises;
generation of excerpts from the subject development plan by main directions, organizations and enterprises in the sector, and of subject plans-orders for co-performing ministries.

The basis of the mathematical model of the planned object is the system of network models of a special type with the specified structure of developments. The structure reflects on the one hand the development stages, and on the other, the component elements of the technical complex being developed.

Example 2. A program package to generate a development subject plan for subdivisions of a KB (NII) [design bureau (scientific research institute)] for a quarter. This package is designed to generate quarterly plans that provide for performing work on a subject by a subdivision, united by the feature of the direction of research or functioning of items. The end product of any theme is a prototype.

The complex of problems for quarterly subject planning for a KB (NII) includes:
calculation of the technical and economic indicators and generation of a draft plan for the quarter being planned;
generation of a report for the last quarter;
generation of a quarterly plan taking into account changes made to the draft by management and information reports;
accumulation of information reports during the quarter and generation of plan supplements at established frequencies.

The basis of the mathematical model for the planned object is the system of linear-network models that defines the structure of the development, the structure of the prototype being developed and the structure of the organizational and financial links between the separate performing agencies.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



Both packages were designed with the aid of the "Vil'nyus" system. In the first package, two of the four processors, and in the second, two of the three processors were designed completely on the basis of the existing problem module library. They effect entry of the initial data and calculation of the time characteristics for the networks. Also developed for the five-year palnning package were modules that distribute resources and generate output documents, and for the quarterly planning package, modules that perform calculation of the plan indicators and generate output documents. In both cases, the system portion was completely built with "Vil'nyus" system facilities.

System Technical Characteristics. The "Vil'nyus" system was developed for the Unified System of Computers; it operates under control of the YeS OS operating system with capabilities of at least that of version 4.1.MVT without subtasks. For system operation, 100K bytes of main storage and about 7M bytes of disk storage are required. Storage needed for execution of the PPP designed is a function of the features of the package itself. The resident part of the system in the process requires 4K bytes of main storage. The system nucleus is programmed in Assembler.

Prospects for System Development. System use has shown that the design stage requires the greatest efforts. Design of a system according to a structure designed in advance and modification of it are then performed relatively easily, although insufficient universality of some tools has been noted, in particular the facilities for data base loading and update.

Further development of the "Vil'nyus" system is planned in the following directions:
development of new problem modules for more complete functional coverage of the subject area;
expansion of the capabilities of existing modules to offer the user a larger spectrum of capabilities for specification of networks (including introduction of the "operation-arc" type network, use of analogs, standards, etc.);
expansion of the problem-oriented macro facilities by facilities based on the concept of abstract types of data;
development of specialized facilities for generation of output documents;
development of interactive facilities for generating and optimizing network models and step-by-step improvment of plans.

BIBLIOGRAPHY

1. Tamm, B and Tyugu, E., "Development of Problem-Oriented Software," KIBERNETIKA, No 4, 1975, pp 76-85.
2. Musstopf, G.; Orlowski, H. and Tamm, B., "Program Generators for Process Control Application," The 2nd IFAC/IFIP Symposium on Software for Computer Control, SOCOCO '79, Preprints, Vol 1, General Computing Center of Czechoslovak Academy of Sciences, Prague, IP-III, pp 1-12.
3. Red'ko, V and Yuzhenko, Ye., "Algorithmic Languages and Translating Systems," KIBERNETIKA, No 5, 1967, pp 87-91.
4. Matulis, V. and Chaplinskas, A., "Structure of Program Package for Calendar Subject Planning System," in "Struktura i organizatsiya paketov prikladnykh programm" [Structure and Organizations of Application Program Packages], Tbilisi, 1976, pp 117-119.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



5. Vaychyulis, B.; Matulis, V.; Sinyuvene, R. and Chaplinskas, A., "Some Problems in Designing Application Program Packages," in "Avtomatizatsiya protsessov planirovaniya i upravleniya" [Automation of Planning and Management Processes], Vil'nyus, No 3, 1976, pp 9-47.

6. Tonkikh, V.; Chaplinskas, A.; and Yuozevichute, G., "Organization of Data in the 'Vil'nyus' System," in "Automation of Planning and Management Processes, No 6, 1978, pp 11-26, Vil'nyus.

7. Vaychyulis, B. and Matulis, V., "Coding and Machine Calculation of Time Characteristics of Large Networks," NAUCH. TR. VYSSHIKH UCHEBNYKH ZAVEDENIY LITOVSKOY SSR. AVTOMATKIA I VYCHISLITEL'NAYA TEKHNIKA, III, 1970, pp 225-234.

COPYRIGHT: Izdatel'stvo "Finansy i statistika", 1981

8545
CSO: 8144/0138



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



# DIGITAL SIMULATION OF CONTINUOUS PROCESSES ON UNIFIED SYSTEM COMPUTERS

Moscow VYCHISLITEL'NAYA TEKHNIKA SOTSIALISTICHESKIKH STRAN in Russian No 9, 1981 (signed to press 21 May 81) pp 58-65

[Article by Ya. Yazhembek, Master of Science, Polish People's Republic]

[Text] Modern programming languages and continuous process simulation systems are used extensively for scientific and industrial research. Digital simulation of the actual processes yields more accurate results than analog. Therefore, many modern general-purpose computers are equipped with a special simulation apparatus. Until recently, Unified System computers had no special software for simulation, and the only capability was implementation of simulation algorithms in a high-level language (FORTRAN, PL/1). However, this requires a good knowledge of these languages and much experience in programming.

The DIANA programming language along with the well-known CSMP language fills this gap in Unified System software operating under control of DOS or OS.

General Description of DIANA. DIANA is a problem-oriented language providing the capability of digital simulation on Unified System computers with the YeS DOS operating system. It makes it possible to solve problems associated with simulation of arbitrary systems described by differential equations or an analog block-diagram. A general diagram of the functioning of the DIANA language and simulation system is shown in fig. 1.

A DIANA program is block-oriented, i.e. developed on the basis of a block diagram that represents a model of a real system. The programmer, by assigning to each block in the diagram a statement that defines the action of this block (integration, delay) and a name that represents the output signal of this block, can describe the structure of a system by using so-called structural instructions. For example, the instruction Y = (INTEG X, X0) executes integration of the equation $\frac{dy}{dx} = x(t)$ with the initial conditions of $x(t = 0) = x_0$.

The set of structural instructions is used as the basis for defining the sequence of calculations according to the corresponding algorithm. A DIANA program is made up of several parts separated from each other by service records. The simulation algorithm is placed between the PROGRAM and END records and can consist of three areas: initial, dynamic, terminal. The initial area is a series of FORTRAN instructions bounded by the INITIAL and END records. The instructions in this area are



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



executed prior to the start of the simulation process and are needed for calculations associated with the definition of values of parameters in the dynamic area, initial conditions and gain factors.

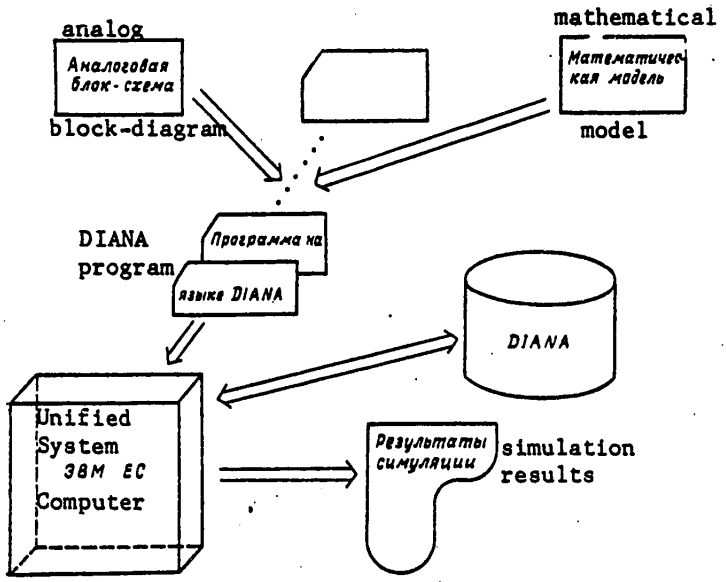



Fig. 1. DIANA system functional diagram

The dynamic area is bounded by the DYNAMIC and END records; it is a series of instructions that describe the simulation process. This area contains the program that executes calculations for one step of integration. At the start of the area, the programmer can define his own structural blocks by describing them as MACRO.

The terminal area is a series of instructions bounded by the TERMINAL and END records; it is executed after completion of each computing cycle, i.e. a slice of time, after which the program can report results through an IO device. In this area, the programmer can change the parameters for a multiple pass of the program or testing of certain conditions.

FORTRAN instructions may also be placed before the initial area and after the terminal area to have them executed before and after, respectively, the simulation process. Here is a general view of a DIANA program:



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



```
F1 { PROGRAM
     SUBROUTINE P1...
     .
     .
     .
     RETURN
     END
     .
     .
     .
     другие команды Фортрана
     other FORTRAN instructions
     .
     .
     INITIAL
     IC1
     END
     DYNAMIC
     MACRO
     .
     .
     .
     END
     IS
     END
     TERMINAL
     IC2
     END
     F2
     GO
     STOP DIANA
```

Note. The F1 area can also contain subroutines in the form of SUBROUTINE or FUNCTION. The IC1 record is a set of instructions that defines the initial conditions. IS is a series of structural instructions. IC2 defines the terminal conditions. F2 is a series of instructions executed after completion of the simulation process. GO is a directive that defines the start of the simulation process.

Structural Instructions. The basic structural block used in DIANA programming is the INTEG integrating block that has as the input variable the name of the signal subject to integration and the initial conditions, and as the output variable, the name of the signal after the integration operation. The structural instruction

Y = INTEG (X, WP)

corresponds to the formula

$$Y(t)=WP+\int_{t_{min}}^{t} X(t)\,dt; \quad Y(t=0)=WP.$$

In this formula, t denotes the independent variable TIME. This structural statement has no arguments and denotes time as an independent variable during the simulation process. Associated with the INTEG statement is the ALGORITHM directive that allows selecting the integration method: either Adams fourth-order method or the Runge-Kutta fourth-order method. The Runge-Kutta method is standard.

In addition to the integration statement, DIANA includes the following functional blocks:

| | |
|---|---|
| DELAY | delay of integration by a step |
| HYST | hysteresis (? of people?) |
| DER | differentiation block |
| RST | RS-trigger |



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



| | |
|---|---|
| BLOCK | isolation block |
| RANDOM | noise generator |
| DSPLACE | dead zone (insensitivity) |
| LIMIT | limiter |
| FG | arbitrary function generator |
| AND | operator for logical multiplication |
| NAND | negation of logical multiplication |
| IOR | operator for logical sum |
| NOR | negation of logical sum |
| NOT | negation of signal |
| COMPARE | comparison of signals |
| EOR | non-equivalence |
| EQUIV | equivalence |
| QNTZR | quantizer operator |
| SINE | generator of harmonic oscillations |
| INSW | input switch |
| FCNSW | functional switch |
| RAMP | ramp function |
| STEP | step function generator |

In addition to these functional blocks, the DIANA language also permits all standard FORTRAN language functions.

Translation and Simulation Results. During translation, a DIANA program is translated into a set of interrelated procedures in the FORTRAN language that execute the structural instructions listed in the program. Then processing is performed by the FORTRAN translator and the simulation process is executed. Upon completion of the latter, results are printed out in tabular or graphic form. The programmer obtains a table of values of variables as a function of time or a graph of a certain variable as a function of time. The phases of processing of the source program, starting with job definition to obtaining simulation results, are shown in fig. 2.

Let us take the following problem as an example.

Example. Let there be a system of chemical reactions of the first order, the constants of which are K1 and K2, respectively:

$$A \xrightarrow{K1} B \xrightarrow{K2} C$$

Let $(A_0)$ be the initial value of factor A at time t=0; let us assume $(B_0)=(C_0)=0$. Then the system of differential equations that represents these reactions takes the form

$$\frac{dx}{dt} = -K1 \cdot x; \quad \frac{dy}{dt} = K1 \cdot x - K2 \cdot y; \quad \frac{dz}{dt} = K2 \cdot y,$$

where $x=(A)/(A_0)$;
$y=(B)/(A_0)$;
$z=(C)/(A_0)$

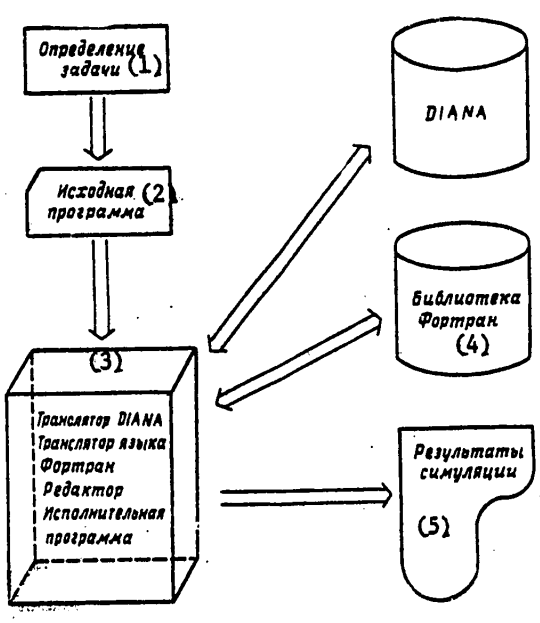
denote the relative values of factors A, B and C, respectively.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6






Key:
1. problem definition
2. source program
3. DIANA translator
   FORTRAN translator
   editor
   executive routine
4. FORTRAN library
5. simulation results

Fig. 2. Source program processing phases

Analytic integration of this system yields the solution

$$x(t) = \exp(-K1 \cdot t);$$

$$y(t) = \frac{K1}{K2 - K1}[\exp(-K1 \cdot t) - \exp(-K2 \cdot t)];$$

$$z(t) = \frac{-K2}{K2 - K1}\exp(-K1 \cdot t) + \frac{K1}{K2 - K1}\exp(-K2 \cdot t).$$

The DIANA program for this example is given in fig. 3. Figure 4 illustrates the results of simulating this problem for factor A.

DIANA System Operating Experience. The DIANA system for simulation of continuous processes was developed and is in use at the Institute of Control Systems in Katowice (Polish People's Republic); it is also in use at the Institute of Mathematical Machines in Warsaw. System operating experience allows making some comparisons. The DIANA simulation system is similar to the CSMP/360 system. The time for executing similar simulation algorithms and the accuracy of results obtained are about identical. DIANA language functions correspond to CSMP language properties, but the choices in method of calculating the corresponding functions are limited. Thus, DIANA offers two methods of integration, the Adams and the Runge-Kutta. In addition to these, CSMP offers the methods of Simpson, Milne, trapezoids and rectangles. The main advantage of the DIANA system is the capability of using it with small computer configurations with 128K bytes of storage (system size is about 108K bytes). More than 128K bytes of main storage and OS are needed for the CSMP system.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



```
PROGRAM
INITIAL
C1=SK1/(SK2-SK1)
C2=-SK2/(SK2-SK1)
END
DYNAMIC
XPRIM=-SK1*X
X=INTEG(XPRIM;X0)
YPRIM=SK1*X-SK2*Y
Y=INTEG(YPRIM;Y0)
ZPRIM=SK2*Y
Z=INTEG(ZPRIM;Z0)
ROB1=-SK1*TIME
XP=EXP(ROB1)
ROB2=-SK2*TIME
ROB3=EXP(ROB2)
ROB4=XP-ROB3
YP=C1*ROB4
ZP=ONE+C2*XP+C1*ROB3
END
END
TMING0.0)
TMAX(5.0)
TSTEP(0.1)
NSTEPS(1)
ALGORITHM(RK)
DATA(X0=1.,Y0=0.,Z0=0.,SK1=2.0,SK2=1.0)
TITLE(1H1,40X,26HWYNIKI PROCESU MODELOWANIA////)
PRINT(X,XP,Y,YP,Z,ZP)
PRTPLOT(X,XP,Y,YP,Z,ZP)
GO
STOP DIANA
```

Fig. 3. DIANA simulation program

```
  TIME          X          MINIMUM= 0.454074E-04                                                    MAXIMUM= 0.100000E 01
0.0          0.100000E 01  I----------------------------------------------------------------------------------------*
0.10000E 00  0.818734E 00  I--------------------------------------------------------------------------------*
0.20000E 00  0.670325E 00  I-----------------------------------------------------------------*
0.30000E 00  0.548817E 00  I--------------------------------------------------------*
0.40000E 00  0.449335E 00  I-----------------------------------------------*
0.50000E 00  0.367886E 00  I--------------------------------------*
0.60000E 00  0.301200E 00  I-----------------------------*
0.70000E 00  0.246603E 00  I--------------------*
0.80000E 00  0.201902E 00  I----------------*
0.90000E 00  0.165304E 00  I-------------*
0.10000E 01  0.133340E 00  I----------*
0.11000E 01  0.110807E 00  I---------*
0.12000E 01  0.907215E-01  I-------*
0.13000E 01  0.742767E-01  I------*
0.14000E 01  0.608129E-01  I----*
0.15000E 01  0.497893E-01  I---*
0.16000E 01  0.407644E-01  I---*
0.17000E 01  0.333751E-01  I--*
0.18000E 01  0.273253E-01  I-*
0.19000E 01  0.223722E-01  I-*
0.20000E 01  0.183169E-01  I-*
0.21000E 01  0.149906E-01  I*
0.22000E 01  0.122782E-01  I*
0.23000E 01  0.100526E-01  I*
0.24000E 01  0.823040E-02  I*
0.25000E 01  0.673850E-02  I*
0.26000E 01  0.551704E-02  *
0.27000E 01  0.451698E-02  *
0.28000E 01  0.369821E-02  *
0.29000E 01  0.302784E-02  *
0.30000E 01  0.247900E-02  *
0.31000E 01  0.202964E-02  *
0.32000E 01  0.166173E-02  *
0.33000E 01  0.136052E-02  *
0.34000E 01  0.111390E-02  *
0.35000E 01  0.911988E-03  *
0.36000E 01  0.746675E-03  *
0.37000E 01  0.611328E-03  *
0.38000E 01  0.500514E-03  *
0.39000E 01  0.409788E-03  *
0.40000E 01  0.335507E-03  *
0.41000E 01  0.274691E-03  *
0.42000E 01  0.224895E-03  *
0.43000E 01  0.184132E-03  *
0.44000E 01  0.150755E-03  *
0.45000E 01  0.123428E-03  *
0.46000E 01  0.101055E-03  *
0.47000E 01  0.827369E-04  *
0.48000E 01  0.677395E-04  *
0.49000E 01  0.554606E-04  *
0.50000E 01  0.454074E-04  *
```

Fig. 4. Simulation results for factor A





APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



The structural and dynamic capabilities of simulation with the DIANA language are similar to the CSSL language. The simplicity and ease of learning the DIANA language extend considerably the group of system users.

Main characteristics of the language are:

simple and clear form of expressions that describe the model;

simple set of statements that allow description of the majority of dynamic problems represented by differential equations;

execution of a process for a group of changing parameters;

automatic sorting of operations;

result output format that allows rapid judgment of the properties of the modeled system; and

the capability of supplementing the language with new functional blocks in the form of MACRO, as well as the use of FORTRAN functions and subroutines.

This programming language can be used in various fields of science and technology: in scientific research experiments, design, for modeling operation of designs and devices, in industrial and economic planning to improve methods of organization and production technology.

Operating experience confirms the great efficiency of the DIANA language and system in solving scientific and technical problems by the method of mathematical models of the real processes, continuous in time.

COPYRIGHT: Izdatel'stvo "Finansy i statistika", 1981

8545
CSO: 8144/0138



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



# PROBLEM-ORIENTED LANGUAGE FOR SYSTEM OF COMPUTER-AIDED DESIGN OF MANUFACTURING PROCESSES OF MECHANICAL MACHINING OF PARTS

Moscow VYCHISLITEL'NAYA TEKHNIKA SOTSIALISTICHESKIKH STRAN in Russian No 9, 1981 (signed to press 21 May 81) pp 65-73

[Article by V. D. Tsvetkov, candidate of engineering science (USSR), A. A. Tolkachev, candidate of engineering science (USSR), and I. Cher, engineer (Hungarian People's Republic)]

[Text] Computers are being used ever more extensively in developing automated systems for technological preparation of production. Used as source information for such systems are the part drawing and manufacturing instructions. At the current level of development of computer technology, the information contained in drawings cannot be interpreted directly by a computer. Therefore, a basic problem in developing an automated system is the development of a formalized language to describe the source and intermediate information in a form convenient for input into a computer, machine processing and output of design results in the form of customary manufacturing documents.

There are a number of languages and methods of coding that permit description of the design and manufacturing characteristics of a part, such as the material, mass, dimensions, qualitative and quantitative parameters of the surfaces to be processed, coordinates of their arrangement in space and others [1-4]. But methods for describing the function and structure of items and the manufacturing processes are still inadequately developed. In contrast to existing systems for coding and the problem-oriented languages, the suggested language contains advanced facilities for describing not only the parameters of the surfaces and the part as a whole, but also the shaping, dimensional and precision relationships of the elements of a part of various levels of complexity. As algorithmic developmental experience shows, it is precisely these relationships that determine the choice of manufacturing solutions.

The language is based on the methodology of the systems approach to constructing the objects and processes of design. Any machine building part is considered a complex part that in accordance with its structure can be broken down into a specified number of simpler interrelated elements.

Surfaces to be machined are taken as the base structural elements of a part, since they are directly related to the primary structural elements of the manufacturing process, the steps. These surfaces are subdivided into three types:



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



elementary (flat, cylindrical, conic and others);
type, consisting of frequently encountered combinations of several elementary surfaces and to be processed by a type tool (shoulders, slots, grooves and others);
normalized, the shape and dimensions of which are prescribed by corresponding standards (center holes, slotted surfaces and others).

In accordance with this, the first level of formalized description is designed to describe shapes, dimensions, precision of machining and the physicomechanical properties of the surfaces listed above.

Description of the Basic Structural Elements of a Part (First Level). The majority of surfaces on machine parts are kinematic, i.e. they can be derived by the motion of a line (generatrix) in space according to a certain law (directrices). The shape of the most widespread elementary, type and normalized surfaces will be specified when there are given the axis of the surface i, the directrices m and the generatrices l of the line: $\Phi$ (i,m,l).

Various types of surfaces result from the great variety of shapes of generatrices with a comparatively small number of laws of motion of these lines in space. In connection with this, the classification glossary of types of basic surfaces is based on the following features: the law of motion in space of a directrix that defines the class of surface (plane, surface of revolution, helical, etc.); the shape of the directrix that establishes the subclass (plane, shoulder, slot, etc.) and the type of surface (rectangular, angular, radius, etc. groove); structural varieties of the surface by length (open, semi-open, closed) (figs. 1, 2). These features determine the design and manufacturing peculiarities of surfaces and affect to a considerable extent the choice of machining method and type of cutting tool.

The description of the shape of surfaces is a mnemonic designation consisting of letters and numbers. The surface class and subclass is designated by a short title in the Russian language, and the type and variety by numbers; for example, a rectangular slot with a junction of lateral surfaces on a radius, semi-open in length is a PAZ13; a cylindrical surface semi-open in length is a TsIL1, etc. Internal surfaces are described with the symbol "—" (for example, a through cylindrical hole is a —TsIL1). This description of the shape of part surfaces is more graphic, which accelerates learning the language and facilities use of it. In the process, the complication of machine processing of information is negligible.

Describing dimensions of type and normalized surfaces has certain features due to their shape being specified by generalized code in the language of the first level, and not in the form of a set of elementary surfaces. In connection with this, it is not possible to indicate elements in clear form, the distance between which is specified by a particular dimension. To uniquely describe dimensions, in the classification tables there are drawings with dimension conventions. Each parameter of a surface is characterized by its type and numeric value (for example, DB50, LB40, etc.).

In describing basic surfaces, a determinant is used that is the set of parameters uniquely defining the shape $\Phi$, dimensions, finish of machining and deviation from the proper geometric shape of the surface P and its physicomechanical properties C:

$$NI = \{\Phi, P, C\}.$$



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



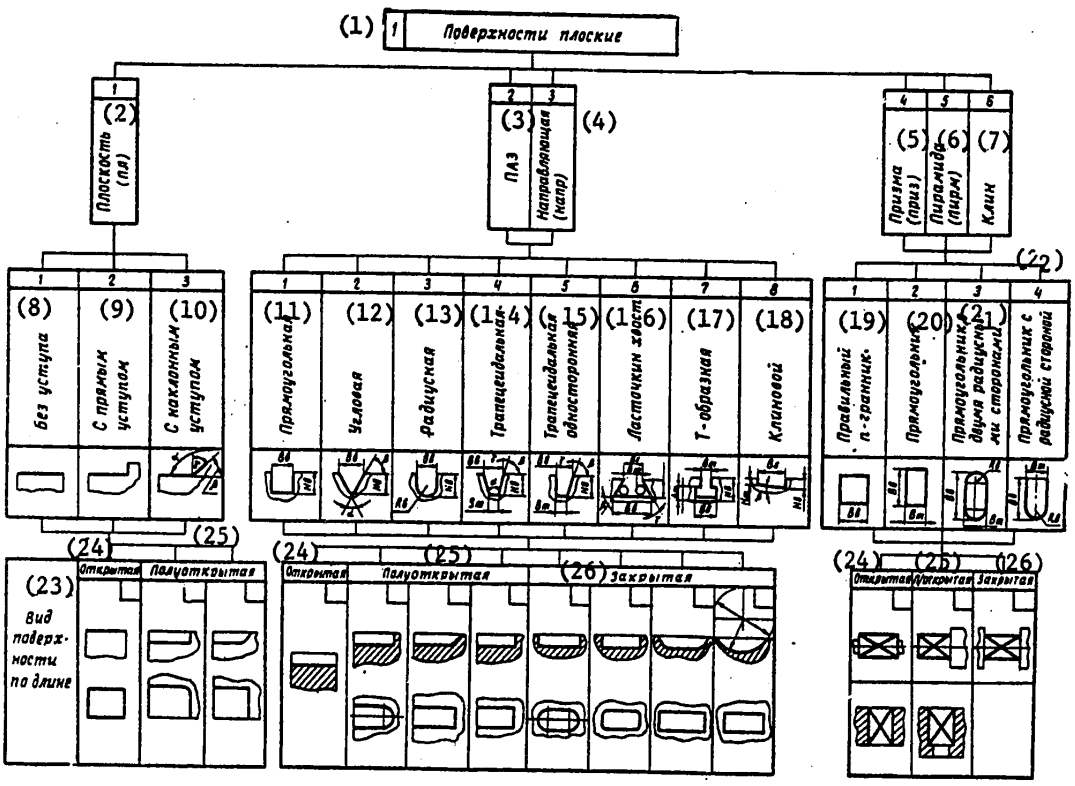



Fig. 1. Diagram of classification and coding for plane surfaces

Key:

1. plane surfaces
2. plane (pl)
3. slot
4. guide (napr)
5. prism (priz)
6. pyramid (pirm)
7. wedge
8. without shoulder
9. with straight shoulder
10. with inclined shoulder
11. rectangular
12. angular
13. radius
14. trapezoidal
15. one-side trapezoidal
16. dovetail
17. T-shaped
18. wedged
19. regular polygon
20. rectangle
21. rectangle with two radius sides
22. rectangle with a radius side
23. type of surface by length
24. open
25. semi-open
26. closed



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



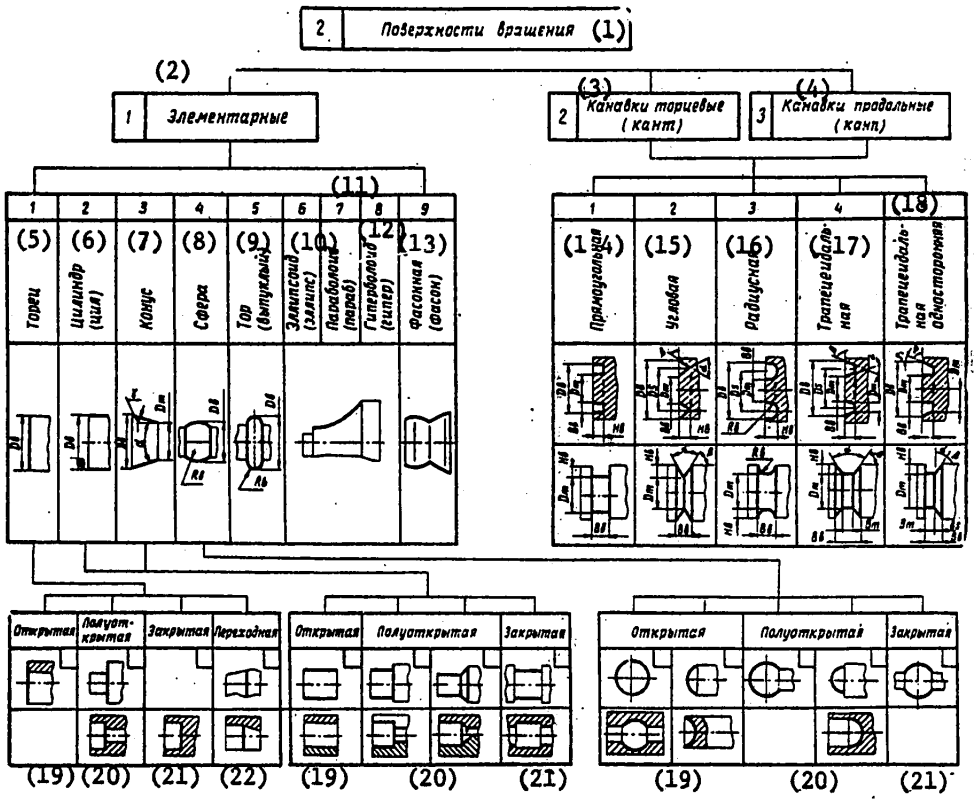



Fig. 2. Diagram of classification and coding for surfaces of revolution

Key:

1. surfaces of revolution
2. elementary
3. face grooves (kant)
4. longitudinal grooves (kanp)
5. face
6. cylinder (tsil)
7. cone
8. sphere
9. torus (convex)
10. ellipsoid (ellips)
11. paraboloid (parab)
12. hyperboloid (giper)
13. contoured (fason)
14. rectangular
15. angular
16. radius
17. trapezoidal
18. one-side trapezoidal
19. open
20. semi-open
21. closed
22. transient



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



A developed structure is built by specification of features that describe the surface parameters.

Let us consider the description of parameters of some surfaces on a part shown in fig. 3:

1 = TORETs1 [face], DB30/FASKA [chamfer], HB3, FG45G;
2 = TsIL2 [cylinder], DB30(H6), RA2.5/GALT [fillet], RB5;
3 = TORETs2, DB40;
4 = TsIL2, DB40 (F7/GALT, RB4).

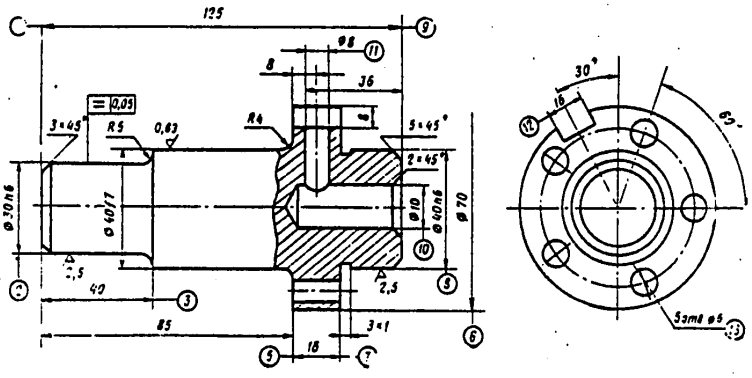



Fig. 3. Splined shaft (numbers in circles are surface numbers)

Description of the Part at the Second Level. Considered at the second level are complex objects (parts or portions of them) formed by combination in space of basic structural elements. This level is intended for description of
the shape of the part;
dimensional relationships between the basic structural elements;
relative position of basic elements (parallelism, alignment and others) and the manufacturing requirements for precision of their relative position (nonparallelism, nonperpendicularity, wobble and others); and
general information on a part or a portion of it at the given level of separation.

A part is considered specified in the rectangular system of coordinates, the axes of which run through the most characteristic elements of the part. For example, for parts of the "body of revolution" type, the system of coordinates is tied to one of the end faces and the longitudinal axis of the part; for plane prismatic parts, the system of coordinates is situated so that the entire part is placed in the first quadrant; for parts with a complex configuration, the coordinate system is tied to the structural bases of the part.

The "union" and "detach" operations are used to describe the shape of a part or spearate portions of it at this level. The union operation for part elements $Q_1$ and $Q_2$ is denoted by the symbol $U_i$ and consists in the design of the more complex object D, derived by joining these elements: $D=Q_1U_iQ_2$.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



The symbol i denotes the nature of the interconnection of the elements in space: $U_1$, by the axis (coaxial); $U_2$, by a right angle; $U_3$, by any non-right angle; $U_4$, tangentially; and $U_5$, by parallel axes of the elements.

The detach operation is denoted by the symbol $\setminus_i$ and describes the complete object derived as a result of detaching the element $Q_2$ from the volumetric element $Q_1$:

$$D = Q_1 \setminus_i Q_2.$$

Just as in the preceding case, the symbol i denotes the nature of the detach operation: by the axis (coaxial) is denoted by $\setminus_1$; by a right angle, $\setminus_2$; by an arbitrary angle, $\setminus_3$; and detaching with parallel axes of elements, $\setminus_4$.

The nature of mutual intersection of part elements $Q_1$ and $Q_2$ is specified by the "intersection" relationship.

The variety of design operations and relationships describe major features of the part shape that are grasped only when looking at the drawing, without specification of which the proper design and manufacturing solution could not be chosen with computer-aided design.

The nature of the relative position of the basic surfaces in the part configuration is specified by the following relationships: coaxiality, $\varepsilon_1$; perpendicularity, $\varepsilon_3$; parallelism, $\varepsilon_4$; symmetry, $\varepsilon_5$; arrangement of elements on a straight line, $\varepsilon_7$; arrangement of elements in a circle, $\varepsilon_8$; and membership (incidence), $\varepsilon_9$. For example, the relationship of parallelism of surfaces $Q_1$ and $Q_2$ is described in the form of $Q_1 \varepsilon_4 Q_2$, and symmetry of surfaces $Q_1$ and $Q_2$ relative to the axis OX by $Q_1 \varepsilon_5 (OX) Q_2$.

The shape of a part as a whole or its individual portions in the language of the second level is described by the graph $\Phi = (X,E)$, in which the vertices correspond to the basic surfaces and the arcs to the operations and relationships that describe the interrelationship of these surfaces. The graph can be represented element by element in the form of relationship tables. Recorded in the first and second columns of the table are the incident vertices, and in the third column, their interrelated edges. Table 1 shows the description of the shape of a part fragment shown in fig. 3. Information on the part shape, written in the form of an element relationship table, is convenient for entry into a computer and subsequent analysis of it in computer-aided design.

Table 1

| i-th element<br>*i-й элемент* | j-th element<br>*j-й элемент* | element relationship<br>*Связь элементов* |
|---|---|---|
| 5 | 6 | $V_1$ |
| 6 | 12 | $\setminus_5$ |
| 12 | 11 | $\setminus_2$ |
| 6 | 7 | $V_1$ |
| 11 | 10 | $\Pi_2$ |



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



The dimensional link between the two elements $Q_i$ and $Q_j$, which may be points, lines and surfaces, is specified by the type of dimension $\mu$, its numeric magnitude b and is described by the relationship $Q_i \mu b Q_j$.

The letters x, y and z denote linear dimensions parallel to the corresponding coordinate axes, and the letter l, those arbitrarily positioned in space. Angular dimensions are denoted by the letter $\phi$, and diametric by D or R. For example, $Q_i X 50 Q_j$ designates a distance of 50 mm along the X axis between the elements $Q_i$ and $Q_j$.

Mathematically, any dimensional chain is described by a graph in which the vertices correspond to part elements (points, lines and surfaces), and the arcs to the type and numeric magnitude of the dimension between these elements. The graph for dimensional links, just as the graph for the part shape, is represented in the form of a table of relationships.

Technical requirements for the precision of the relative position of the part surfaces are defined by the type of deviation $EH_i$ and its numeric magnitude $\delta$ and are described by the expression $Q_i EH_i \delta Q_j$.

The following designate deviation parameters: $EH_1$, misalignment; $EH_2$, wobble; $EH_3$, non-perpendicularity; $EH_4$, non-parallelism; $EH_5$, asymmetry; and $EH_6$, nonplanarity. For example, a wobble of surfaces $Q_1$ and $Q_2$ relative to the surface Q of no more than 0.05 is notated by the relationship $(Q_1, Q_2) EH_2 \leq 0.05 Q_3$.

The set of technical requirements for the precision of the relative position, just as the linear dimensional chains, is described by a graph, the vertices of which correspond to the part elements, and the arcs to the type and numeric magnitude of the deviation parameters. This graph, just as in the preceding cases, is represented in the form of a table of relationships. Table 2 gives the description of the dimensional relationships and technical requirements for precision of the relative position of the individual surfaces of the part shown in fig. 3.

Table 2

| i-th element / i-й элемент | j-th element / j-й элемент | element relationship | |
|---|---|---|---|
| | | type | dimension |
| 1 | 3 | X | 40 |
| — | 5 | X | 85 |
| — | 9 | X | 125 |
| 2 | OX | EH4 | 0.05 |

The first three lines in the table describe a cluster of linear dimensions specified from surface 1, and the fourth, the nonparallelism of surface 2 to axis OX.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



General information on the part or its individual elements are characterized by the set of parameters

$$\Phi D=\{HD, ND, LB, BB, DB, HB, MD, TO, SB, RA, RZ, GD, \Phi H, DH\},$$

which describe the designation HD and number ND of the part; its dimensions LB, BB, DB and HB; the material MD; heat treatment TO and mechanical properties SB; the general treatment finish RA and RZ; weight GD, and magnitudes of chamfers not specified in the drawing ΦH; and nonspecified maximum deviations DH.

The overall formalized description of the part includes general information on the part, complete information on the shaping, dimensional and precision relationships of the part elements between each other and the parameters of each elementary, type and normalized surface to be treated. For some problems in manufacturing design, only part of this information is used; in this case, an abbreviated formalized description of the parts to be treated is compiled and the separate levels are integrated for convenient initial description of the parts.

BIBLIOGRAPHY

1. Begunov, G. S. et al., "Apparatus of Geometric Descriptions for Design Automation," in "Vychislitel'naya tekhnika v mashinostroyenii" [Computer Technology in Machine Building], Minsk, Izd-vo AN BSSR [Belorussian SSR Academy of Sciences], 1966.

2. Goranskiy, G, K., "Sistema kodirovaniya informatsii pri mashinostroitel'nom proyektirovanii" [Information Coding System for Machine Building Design], Minsk, Izd-vo AN BSSR, No 1, 1965.

3. Nikitenko, V. D., "Podgotovka programm dlya stankov s programmnym upravleniyem" [Preparation of Programs for Numeric-Controlled Machine Tools], Moscow, Mashinostroyeniye, 1973.

4. Brown, S. A. et al., "Description of the APT Language," COMM. OF THE ACM, No 11, 6, 1963.

COPYRIGHT: Izdatel'stvo "Finansy i statistika", 1981

8545
CSO: 8144/0138



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



EXPERIENCE IN USE OF TIME SHARING SYSTEM FOR PROGRAM DEVELOPMENT

Moscow VYCHISLITEL'NAYA TEKHNIKA SOTSIALISTICHESKIKH STRAN in Russian No 9, 1981 (signed to press 21 May 81) pp 73-81

[Article by G. Brusdeylins, engineer (GDR)]

[Text] Unified System data processing systems are used in two modes: batched and interactive. Under batched processing, a job is fully described before being sent for processing. The jobs thus prepared form a continuous stream of entry data to be processed with the aim of optimal loading on the data processing system. After job entry, the user may not affect the course of its execution and intervene in it, i.e. does not take part in the processing process. The time of processing is determined by the computer center. As a result, the user waits a long time for his job and this is typical for this mode. Waiting time is calculated from the time the user submits his job until he receives results. In the interactive mode, jobs are processed in steps by entry of partial jobs and output of responses. Continual user-system communication is established by using a terminal.

The interactive mode is subdivided into the mode of shared use with access to all resources and the mode of shared use with access to specified (limited) resources. The latter allows processing from a terminal of a set of tasks specified by the system (for example, seat reservations, literature search). The volume of tasks is not restricted by the system so that various subscribers may solve all problems independently of each other. The mode of shared access to all installation resources can be effected by using the components of the YeS OS "Time Sharing System" (SRV) [TSS].

TSS is an extension of the YeS OS operating system. TSS is based on the control program MVT, and in future, SVS will be the basis.

TSS and the batched mode are compatible since in the process the same operating system components are used. The data formats, notation, access methods and compiling and service programs match in both modes. Thus, programs intended for batched processing when developed can be run under TSS without change. However, using the BTAM [basic telecommunications] and GAM [graphic] access methods and their EXCP [execute channel program] equivalents is not possible. A minimum of 512K bytes is required to place TSS in main storage. Simultaneous operation in TSS and the batched mode is possible.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



The start of the operation on a terminal is called the start of a session. A subscriber communicates with the system by using the TSS command language. By using the command LOGON, the subscriber establishes communication with the system, indicating his identifier and procedure name (the procedure contains a job control statement). This procedure, supplemented by certain TSS data, describes the TSS job that is included in the operation until the end of the session.

Tasks, executed at the subscriber's option, are described by entry of commands. Then the appropriate command processor is loaded into the TSS area, assigned to the subscriber and executed. Missing data are requested by the system. The commands are used to check the course of the session and control it, to compile, execute and debug programs, to manage data and to manage and check the system. If a sequence of commands is needed repeatedly, it is advisable to store it in a file as a procedure that can be called by using the EXEC command. A session is terminated by the LOGOFF command or by a new LOGON command.

If inclusion of TSS was provided for when the operating system was generated, the TSS mode can be started at any time by the start command. After this, the operator can vary the TSS mode or terminate it by using other commands. This means that TSS is simply adapted by subscribers to varying loads. Since TSS is used for data transmission from and to terminals by an extended remote access method (TCAM), the system must contain the message control program [MCP] that is capable of servicing both TSS and other modes of application for TCAM.

After TSS is started, users are granted one or more storage areas of a specified size for job execution. Remaining main storage can be used to process jobs in the batched mode. Each TSS area processes several TSS jobs, though only one subscriber's job is in this area at any given time. This condition stems from the TSS job staying in this area and being given access to system resources only for a certain time (a time slice). After expiration of the time slice, the contents associated with the job are copied and stored in direct access storage, and another job is written to main storage (OP). This process occurs rapidly; therefore, a subscriber has the impression that he alone is operating with the data processing [DP] system.

TSS Application Areas. TSS is suitable for use in many fields. The main five are considered below.

Application in Science and Technology. This includes use of the computer by engineers, statisticians, mathematicians and other specialists who need not be well versed in DP. In the process, problems occurring in everyday work are solved and this is done by using a terminal installed in the specialist's working area.

Data Processing. TSS permits creation and change of files, source modules of programs, data, texts (program documentation, for example) and procedures.

The capability of changing source modules is especially important in software development. Word processing is assuming great importance.

Interactive Programming. Software development from a terminal in TSS mode is called interactive programming. In the process, there is the capability of developing, modifying and executing programs intended for the batched and TSS modes. The latter offers efficient facilities for program debugging. Programmer productivity can be raised considerably because of interactive programming.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



Job Processing in the Conversational Mode. Generating jobs under TSS and processing them in the batched mode is called job processing in the conversational mode. In the process, conversation is permitted only during job generation and when results are received. Processing occurs without subscriber intervention. Job processing in the conversational mode should be used in compiling programs when the programs process a large quantity of data and execution time is very long; there is no conversation during processing (during translation of programs, for example); and
when special technical data media have to be installed.

Interactive Application. Problems, as they are realized with use of a computer with access to particular resources, are also solved in the TSS mode (for example, making information available, current accounting of availability, data acquisition).

TSS Application Example. The scientific research center of the Robotron Scientific Production Combine is developing problem-oriented software for general-purpose use. The DBS/R data base management system is typical of these software facilities.* TSS is provided to more efficiently develop such software systems. A TSS test by a limited number of programmers was run for a year to gain experience. Results were used to define agreements for the TSS mode and the hardware composition; the number of programmers included in the operation was expanded.

A YeS-1040 installation with 1024K bytes of main storage was used to try out TSS. Two hard communications channels were used; a YeS-8564 remote display unit with nine screens was connected to each channel. At the start, TSS was used six hours a day. The installation was operated under control of YeS OS/MVT 6.0 and was available primarily for work on software. The installation also operated intensively in the batched mode. Under these conditions, main storage was allocated as follows:

| | |
|---|---|
| 102K | system area for resident load modules; message scheduler |
| 64K | message control program (TCAM); TCAM buffer storage |
| 96K | TSS control area; TSS buffer storage; resident load module area |
| 130K | TSS area; (10K) local system queue area |
| 482K | batched processing area |
| 150K | system queue area; nucleus of control programs |

Software Development. Prior to the introduction of TSS, all jobs for software development were executed during the day by groups of developers at the computer center according to a general work process. At night, this work was done under closed shop conditions. After TSS was introduced, the following work was accomplished preferably under TSS:
modification of source modules;
change of data, cataloged procedures and command procedures;
debugging of individual programs;
translation of small programs; large source modules were translated under batched processing;

---

* Bittner, Yu. and Guenther, M., "DBS/R Data Base Management System for Subsystems in the 'Manpower' System," in "Vychislitel'naya tekhnika sotsialisticheskikh stran" [Computer Technology of the Socialist Countries], No 5, Moscow, Statistika, 1979.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



work on maintenance of program libraries; and
word processing (composition of documentation).

The batched mode includes:
overall debugging of software systems;
debugging of programs with a large quantity of data;
translation of large programs;
operation with magnetic tapes;
compilation of source modules; and
execution of debugged programs.

Operation at night continues under closed shop conditions. This method assumes compatibility of data and results in the TSS and batched modes. Also, the approach was to use the same program libraries in both modes. TSS supports this compatibility.

Management of External Storage Units. Availability of high-capacity external storage units is assumed in the TSS and batched modes to provide access to the libraries and data needed by many programmers. To reduce this requirement to the minimum, all required storage was divided into that with continuous and that with non-continuous access. Storage with continuous access included the libraries and data often used by the programmer and which, therefore, had to be available during the entire computation. Parts of storage referred to infrequently by the programmer were allocated only at a certain time; recorded there was information rarely needed, for example, source modules of debugged programs. All programmers use libraries that are made available under a common principle (agreements on names, estimate of quantities). This principle permits getting away from personal libraries for subscribers and provides that about five to ten programmers for a software system use common libraries in both the TSS and batched modes.

To develop Assembler software, the following system of subdivided files is made available: library for source modules under development; library for load modules; procedure library which includes command procedures and cataloged procedures for batched mode; and a library to hold in-house macro instructions. Use of external storage is observed from a centralized service console. Subscribers may place temporary files in main storage.

Program Compilation. Programs are compiled as follows. Statements are written on forms and punched. Initial translation is performed in the batched mode and the source module is kept as a component of a subdivided file. If errors occur in translation, the printout is used to eliminate them in the TSS mode. The EDIT command is used to correct errors detected, and the program is then retranslated. When the source modules are large, this can be done by sending the job for batched processing. After its execution, the subscriber receives a message (he engages in other work in the meantime) and then checks the diagnostic message file by using the OUTPUT command. Small source modules are translated under TSS. In the process, diagnostic messages are output directly to the terminal. Errors are corrected by using the EDIT command. The printout is used to debug the program. After the files needed for program execution are allocated by the ALLOCATE command, the program to be debugged is executed by the TEST command. After errors are detected and corrected, thk program is sent for overall testing if it is a component of a software system, or for final testing in the case of separate programs. Final



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



tests are performed in the batched mode. After successfully passing the test, a source module is transferred from the development library to the archive library, and the load module to the production library. Then these programs are deleted in the development libraries.

Enhancement of Programmer Productivity. Programming is performed in the following work stages:
analysis of the problem and draft of the program (formulation of the job and compilation of the program flow chart);
coding (formulation of the problem solution by using programming language statements);
program debugging (preparation of instructions in the source language in a form permitting automatic entry into the computer and storage in files, translation, assembly and debugging of programs, search for and correction of errors);
documentation (compilation of application documentation);
maintenance of programs (changing and elimination of errors in debugged programs (corresponds to the program debugging stage)).

With interactive programming, the highest increase in programmer productivity is achieved at the stage of program debugging and documentation compilation. This is due to the following factors:
reduction in waiting time. The long waiting time for job execution typical for batched mode is considerably reduced in the interactive mode because sending the data to the computer center, waiting for the start of data processing and return of results are eliminated. Reducing waiting time significantly reduces software development time;
continuity of work. For a programmer to work efficiently, it is advisable to have him continuously engaged in software development. The long wait for results of job execution in batched processing forces a programmer to work on several programs at the same time so as to not waste time. Shifting continuously from one problem to another lowers programmer productivity;
continuous access to all system functions from the work place. The continuous access to all system functions from the work place during work time raises a programmer's enthusiasm for work since he can correct an error in a program right away when it is detected. Continuous access is a prerequisite for continuous work. The absence of time spent unproductively and the access to all documents needed improves the programmer's working conditions;
efficient utilities. TSS offers a number of efficient utilities, for example the EDIT command, that are not available in the batched mode. Dynamic debugging is of special importance: in the batched mode, debugging utilities are static, i.e. they are programmed in advance, cannot be changed during debugging and debugging can be terminated only after a complete run. Dynamic debugging facilities allow making changes to programs while they are being debugged using information gained while the program is executing. In the process, runs can be terminated on time or repeated after correction. Experience shows that productivity is raised an average of 50 percent in the program debugging stage; this matches foreign experience. This productivity increase in the most laborious stage of software development should be used as the basis for more extensive use of TSS for this purpose.

Requirements for Organization of TSS. One reason for the productivity increase in interactive programming is the drop in a number of organizational requirements that programmers are forced to meet in batched processing. For example, the DP



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



installation, which is an auxiliary facility, has to be "subordinate" to the programmer's working rhythm, and not vice versa. In the ideal, realization of this requirement means: each programmer has to have his own terminal and receive the optimal services from the system. Such requirements are unacceptable for economic considerations, yet the approach must be to reduce organizational restrictions with respect to the programmer. This requirement determines the organization of the TSS mode.

Experience shows that the duration of a debugging session is one to two hours. If this time is considerably exceeded, work concentration drops off. On the average, one to two sessions are required per day for a programmer. Consequently, one can assume that one programmer works no more than three hours per day at a terminal. During a session, a programmer must not experience psychological pressure from the need of keeping within the time allocated to him, when a terminal is made available to a programmer only according to a plan, at a specific time and for a specific period. This contradicts the requirement to free the programmer to the greatest extent from "organizational pressure." To meet this requirement, one terminal is needed for approximately every three programmers.

A terminal may be installed centrally or decentrally. The centralized installation in a separate room ensures a high load on the hardware and a suitable environment for concentrated work. Also, in furnishing the room with equipment, health norms must be taken into account regarding noise, air conditioning, etc. When the terminal is installed in the work area, less additional space is required, but the useful load on the terminal declines.

Adaptation Problems. After system generation, a small group debugged it using TSS. At this stage, the following work was performed:
a study was made of optimal system design to obtain responses quickly;
technology was defined (agreement on names, time of use, etc.);
standard procedures and commands were developed;
utility programs and working documents for programmers were compiled.

After this, the programmers engaged in interactive programming went through a short training course (about six hours). The course covered the general concepts of the hardware, TSS and assignments of the plan with a phased breakdown, by which the command language could be learned on one's own. Then the first session was held under the direction of experienced specialists. All necessary operations were mastered within 14 days.

Supplementary use of interactive programming along with the conventional method for software development requires the programmer to always be ready to shift from one problem to another. A skilled programmer considers the capability of focusing on the main problems as an enhancement of the prestige of his work. Once a staff member has become familiar with interactive programming facilities, he works more productively, and if these facilities are removed, it is considered a demotion.

TSS use over a year has shown that it is an effective means for making software development more efficient. Therefore, comprehensive expansion of TSS use is planned and a "Series-2" Unified System Computer (the YeS-1055) will be used for this. It is planned to use the YeS-7920 as the terminal.

COPYRIGHT: Izdatel'stvo "Finansy i statistika", 1981

8545
CSO: 8144/0138



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



SZPAK-77 INTEGRATED PROGRAMMING SYSTEM FOR COMPLEX AUTOMATION

Moscow VYCHISLITEL'NAYA TEKHNIKA SOTSIALISTICHESKIKH STRAN in Russian No 9, 1981 (signed to press 21 May 81) pp 81-86

[Article by A. Aderek, Master of Science, Polish People's Republic]

[Text] Progress in digital equipment and the evolution in methods of controlling industrial processes have caused a rapid increase in the number of automated systems with control computers (UVM). At the same time, continuous growth in the cost of software development has been noted. Therefore, digital equipment producers and users are very interested in special programming languages that, on the one hand, would lower the labor input for programming, and on the other, would be easily mastered by specialists without programming experience.

The SZPAK-77 programming system, developed at the Industrial Institute of Automation and Measurements MERA-PIAP, is described in this article.

The SZPAK-77 system is intended for the System of Small Computers.

The experience of application of the preceding SZPAK system in industry has been taken into account in the SZPAK-77 system.

The SZPAK-77 system combines the advantages of a form language and real-time FORTRAN. By its function, SZPAK-77 is a general-purpose basic software, i.e. can be used for any set of hardware.

The main purpose of the system is programming control computers that implement supervisory control in real time in automated process control systems (ASUTP). The SZPAK-77 form programming language in this system is intended for programming typical algorithms for data acquisition and preprocessing and control. Algorithms for processing process variables defined by forms can be expanded by using supplementary programs, SUP. The expanded real-time FORTRAN IV F language is intended for preparing them. Supplementary programs communicate with the so-called special programs, optimization programs for example, that are also written in expanded FORTRAN.

Capabilities of making changes to the system, expanding the system, receiving information about variable processes, etc. are provided by the operator-system interactiev language.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



Process Variables. The basic unit of the form programming language is the process variable (PP) [PV]. It is recorded in the system after introduction of the completed form shown in the figure. The form is modeled on those used in the BICEPS programming system. This form is filled out most often for programming the recording and processing of a parameter measured in an object. In this case, the source of the process variable is the transducer signal fed to a computer by devices for communication with an object. The converted value of this signal and its associated supplementary parameters such as, for example, restrictions on the maximum and minimum, are accessible to user programs and the operator.

Usually in an ASUTP, the values measured in an object are used as the basis for determining certain integrated technical and economic indicators that describe the course of an industrial process. Forms can be filled out for these values too, which then become accessible to the operator by proper names. This is the way values are handled that are obtained, for example, as a result of performing chemical analyses in a laboratory, and the operator inputs them manually.

Thus, a process variable is any value measured automatically in an object, computed by a program or input by an operator, for which a SZPAK-77 system form is defined (filled out).

A process variable is identified in the system by using a name, which consists of four characters in the SZPAK-77 version for the MERA-400 control computer. The first character must be alphabetic and the rest numeric. In the following description, ANNN denotes a process variable name.

Digital Control Loop. PV processing algorithms can be divided into two groups. The first includes algorithms for monitoring filtration, scaling by physical units and checking the bounds of the current value of the PV. As a result of their execution, the current PV value is stored in computer storage. The second group includes processing algorithms programmed in supplementary programs and algorithms for computing the control process. Let us call the set of algorithms in the second group and the characteristic parameters for a given PV the digital control loop (KTsP).

It is especially convenient to single out the digital control loops in the initial stage of system debugging and operation since this provides the capability of debugging the software in stages, debugging in the first place the software for central recording and processing of data, i.e. everything except the digital control loops.

System Variables. Associated with each process variable is a number of characteristic values, for example the current value read out in an object at a given time and limitations on the maximum and minimum. Let us call these values system variables (SP) [SV]. They are stored in the form of real numbers, can be read out and measured by an operator, and are also accessible to the remaining portion of the SZPAK-77 software system.

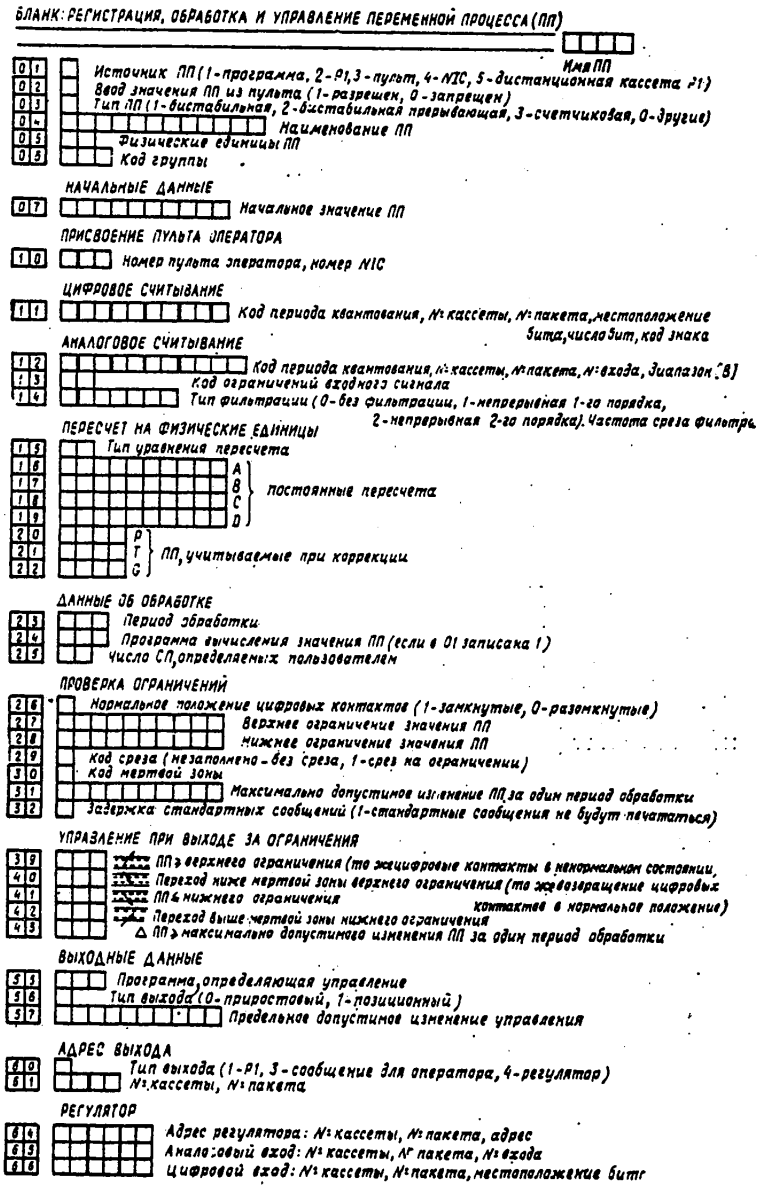
System variables are identified by a name formed from the name of the process variable with which the given SV is associated and two numbers that are the SV number. System variables are listed below.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6






Form for process variable processing algorithms in the SZPAK-77 programming system

[Key on following page; sequential line numbers used as keys]



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



FORM: RECORDING, PROCESSING AND CONTROL OF PROCESS VARIABLE (PV)

PV name

01 - PV source (1, program; 2, R1; 3, console; 4, NIC; 5, R1 remote cassette)
02 - console input of PV value (1, authorized; 2, not authorized)
03 - PV type (1, bistable; 2, bistable interrupting; 3, meter; 0, others)
04 ---------- PV designation
05 -- PV physical units
06 --- group code

INITIAL DATA

07 ---------- PV initial value

OPERATOR CONSOLE ASSIGNMENT

10 --- operator console number, NIC number

DIGITAL READOUT

11 ---------- quantization period code, number of cassette, number of pack, bit location, number of bits sign code

ANALOG READOUT

12 ---------- quantization period code, cassette number, pack number, entry number, range [V]
13 -- input signal restriction code
14 ------- filtration type (0, without filtration; 1, continuous of 1st order; 2, continuous of 2nd order). Filter frequency cut-off

SCALE BY PHYSICAL UNITS

15 -- scale control type
16 ---------- A
17 ---------- B
18 ---------- C scale constants
19 ---------- D
20 ---- P
21 ---- T PV's, taken into account during correction
22 ---- G

DATA ON PROCESSING

23 --- processing period
24 --- PV value computation program (if there is a 1 in 01)
25 -- number of SV's defined by user

LIMIT CHECK

26 - normal position of digital contacts (1, closed; 0, opened)
27 ---------- PV value upper limit
28 ---------- PV value lower limit
29 - cut-off code (blank, no cut-off; 1, cut-off at limit)
30 - dead zone code
31 ---------- maximum permissible change of PV in one processing period
32 - inhibit standard messages (1, standard messages will not be printed)



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



CONTROL WHEN LIMIT EXCEEDED

39 --- PV upper limit (then the digital contacts are in an abnormal state)
40 --- passage below dead zone of upper limit (then digital contacts are returned to normal position)
41 --- PV lower limit
42 --- passage above dead zone of lower limit
43 --- PV maximum permissible change of PV in one processing period

OUTPUT DATA

55 --- program that defines control
56 - type of output (0, incremental; 1, positional)
57 ---------- maximum permissible control variation

OUTPUT ADDRESS

60 - type of output (1, R1; 3, message for operator; 4, controller)
61 ---- cassette number, pack number

CONTROLLER

64 ------ controller address: cassette number, pack number, address
65 ------ analog input: cassette number, pack number, input number
66 ------ digital input: cassette number, pack number, bit location

| System Variable Number | System Variable Designation |
|---|---|
| 00 | current value of PV |
| 03 | limit on maximum of PV (for PV with two states, the normally closed position of the contacts) |
| 04 | limit on minimum of PV |
| 07 | maximum permitted variation in PV value in one processing period |
| 14 | supplementary SV's specified by user |
| . | |
| . | |
| . | |
| . | |
| 13 + N | N is the number of supplementary SV's |

Process Variable State. In a process control system, a computer acquires data and issues control signals incident to many industrial apparatuses that may be faulty or down for maintenance. Some part of the industrial process may not meet required conditions. All this has an effect on the state of process variables.

The current value of a process variable may be within limits, thus proper, or beyond limits, and thus considered doubtful. The current value of a PV may be read out automatically or, in the case of a measuring circuit failure, input manually. A PV may be in a system, but updating of its current value may have ceased, and access to system variables for a user program may be prohibited.

Indicators with two values have been incorporated in the SZPAK form language to allow the capability of sending information on PV states. PV states are listed below.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



| Process Variable Designation | Value |
|---|---|
| PV activeness | 0, active PV; 1, inactive PV |
| digital control loop activeness | 0, active; 1, inactive |
| PV source | 0, PV input automatically; 1, PV input manually |
| PV correctness | 0, correct PV; 1, doubtful PV |
| digital control loop output mode | 0, with control output; 1, without control output |

**Programs Supplementing Processing of Process Variables.** The amount of processing of process variables defined by forms may be expanded by using supplementary programs (SUP) that are executed under conditions specified by forms. These programs are not independent tasks in a system, but are executed as subroutines of the PV processing problem. They are identified by a three-character name. If a subroutine has to be executed under certain conditions, its name must be entered on the appropriate line of the form. Names of supplementary programs are common for a system. Any number of process variables can be called to a program.

Supplementary programs are written in expanded FORTRAN. Expanded FORTRAN, on one hand, provides the capability of executing operations in real time and controlling the operation of a task; on the other hand, it enables access to the SZPAK data base and permits interaction with the remaining parts of the system.

The first group of subroutines is determined by the standard language of real-time FORTRAN. The other subroutines are listed below.

| Subroutine Functions | Call to Subroutine |
|---|---|
| read out values of system variables | CALL FETCH (i, n, w, m) |
| store values of system variables | CALL STORE (i, n, w, m) |
| read out statuses of a process variable | CALL STATUS (i, n, j, m) |
| store statuses of a process variable | CALL ASTATUS (i, n, j, m) |
| process process variable out of turn | CALL SPECIAL (i, n, m) |
| output of controls | CALL OUTPUT (i, n, w, m) |
| printout texts (with buffering) | CALL PRINT (k, l, f, $e_1$, $e_2$, ...) |
| printout texts in ISO-7 code (with buffering) | CALL PRETEXT (l, p, t) |

Note. Values of parameters:
- i number of PV or SV
- n names of PV or SV
- w defines address of REAL type variables where the values read out by the subroutine from the SZPAK data base are to be moved, or from where the values to be stored by the subroutine in the data base are to be moved
- j status words to be read or stored from/to the SZPAK data base
- m INTEGER type variable. After return from the subroutine, contains information on method of execution of subroutine, possible errors, etc.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



k number of parameters
l number of printer
f print format
$e_1$, $e_2$, $e_3$ values to be printed
p number of characters
t text to be printed

Special Programs. In the SZPAK-77 system, a user can program any algorithm, for optimization for example, that uses data acquired and processed by programs prepared in the form language and by supplementary programs. These programs exist in a system as independent tasks and are called special programs. They are programmed in FORTRAN with expansions identical to the expansions for supplementary programs.

Operator-System Interactive Language. An operator can communicate with the system by using the console for the industrial process operator or the DZM-180-KSR serial matrix printer with keyboard. The operator interacts with the system by inputting interactive language directives into the computer according to certain rules. Each directive has its own key on the industrial process operator console. If the message is sent through the DZM-180-KSR, a three-letter code is used for each directive. It is assumed that the basic system monitor is assigned the number 0 and that it is authorized for use in reading and changing all data on any PV and starting the system. Other consoles are assigned sequential numbers starting with 1. They can be used to read all data on any PV, but changes are permitted only for those PV's assigned the number of the given console. Operator-system interactive language directives are listed below.

Code

| | | | |
|---|---|---|---|
| IDN | PV identification | VST | PV status |
| PVR | PV current value | LST | digital control loop status |
| PVH | PV upper limit | PVS | source of current value |
| PRL | maximum permitted variation of PV in one processing period | OUM | control output mode |
| | | EXT | special processing of PV |
| OIL | maximum permitted variation of control value | REP | summary of information on PV |
| | | ALP | PV designation |
| DAT | system variable | PAL | cancel signaling |
| OUT | digital control loop output value | | |

COPYRIGHT: Izdatel'stvo "Finansy i statistika", 1981

8545
CSO: 8144/0138



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



DOS RT REAL-TIME OPERATING SYSTEM

Moscow VYCHISLITEL'NAYA TEKHNIKA SOTSIALISTICHESKIKH STRAN in Russian No 9, 1981 (signed to press 21 May 81) pp 86-90

[Article by A. Shuba, master of engineering, Polish People's Republic]

[Text] In cooperation with the other CEMA countries, the DOS RT real-time operating system has been developed in the Polish People's Republic for the system of small computers (SM EVM).

DOS RT was designed for the SM-3 and SM-4 minicomputers. As is known, the instruction set for the SM-3 and SM-4 processors provides direct addressing of 64K bytes or 32K words (16-bit) of main storage.

The SM-4 processor permits operation with special main storage control units (storage dispatcher) that expand the address range to 256K bytes (128K words). Depending on whether or not there is a need for the storage control unit, it is possible to generate DOS RT versions with or without address translation.

A broad range of peripherals produced in the CEMA countries can be used in the SM-3 and SM-4 systems. These include various types of character terminals and alphanumeric printers with or without storage control; it is possible to generate versions for display, industrial and laboratory interfaces (including the CAMAC interface, for instance). DOS RT permits efficient use of all these devices with full use of their functional hardware capabilities.

DOS RT imposes no major additional restrictions on the field of applications compared to the limits involving hardware capabilities. The structure and programming languages supported make the system most efficient in operating under the following conditions and fields of application:
real-time operation. This application assumes acquisition and processing of measurement data depending on time and control of real and simulated physical processes. DOS RT does not rigidly restrict the number of actions performed concurrently in real time;
software development. The operating system permits development of programs written symbolically, correction of them and translation into a load module; it also provides facilities for program debugging;
data processing. DOS RT permits operation of data base management programs. Such programs provide the capability of creating, updating and managing data files, such as payrolls, inventory reports, etc. The COBOL compiler facilitates development of such programs;



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



numeric calculations. The FORTRAN IV compiler supports the capability of starting and operating programs for engineering and scientific calculations.

The capability of developing networks has a special effect on the use of minicomputers operating under control of DOS RT. Communication interfaces permit connecting minicomputers into a network with an equal-rights or hierarchical structure.

DOS RT may be generated as a system specialized for one or more of the applications mentioned, i.e. it can function in different modes at the same time. This property allows adapting a working version of a DOS RT system to a working configuration of hardware and operating requirements.

DOS RT has an expanded system for operating with files on disk and tape. The tape file structure corresponds to the American standard.

DOS RT fully automates creation, retrieval, expansion and deletion of files. A file owner cannot affect a file and cannot obtain information on the physical location of a file on a disk volume.

The system provides protection of access to user files for operations such as reading, writing, expanding and deleting. Access protection may concern the following categories of users: file owner, user group, system programs and user community.

DOS RT provides the capability of operating with MACRO macroassembler, FORTRAN IV and FORTRAN IV PLUS, CCBOL and BASIC.

Standard system facilities include the MACRO translator and FORTRAN IV compiler. All translators produce an object module. It is common to all languages. This provides the capability of connecting modules written in different source languages at the stage of task development into one task (a task is a program in a form suitable for starting).

BASIC is an exception since it has not only the property of compilation, but can operate in the interpretation mode, i.e. it can directly execute a source program written in a source file or output from a terminal keyboard.

A programmer interested in using all the properties of the SM equipment can use the MACRO macro assembler. This Assembler provides the capability of creating programs consisting of processor instructions and of using macro instructions formulated by the user in a given source module, or in macro system libraries or in a user macro library. Special translator directives offer a number of other facilities: conventional translation, sectioning of a program, segmentation of a program and others.

It is well known that FORTRAN IV is used primarily in numeric computations. Implementation of FORTRAN IV in the DOS RT system also permits using tasks written in this language for real time operation. This is possible because of a library of procedures that organize the time relationship of the task written in FORTRAN and communication with industrial and laboratory interfaces. The FORTRAN IV compiler has a number of expansions compared to the standard language.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



The DOS RT system also includes the FORTRAN IV PLUS optimizing compiler. The efficiency of the code generated by this compiler is two to threefold greater than that obtained from the standard compiler. The shortcoming is its longer compilation time. It can operate only in a system with address translation.

DOS RT has a large set of application programs (programs for editing source files, translation and compilation, building tasks, managing libraries, making changes to various types of files, etc.). Two editing programs are provided in the system:
EDI is an interactive editing program used to create and correct any text file, including source programs. One can use the standard instruction set and create macro instructions formed from combining standard instructions. The program permits editing lines and blocks and supports search and context changes;
SLP is batch editing program used to create and correct and text file by using a series of instructions written in advance in a file. Interactive operation is possible.

During system operation and development of many programs, a considerable portion of program modules may be used repeatedly in various programs. It is advisable to store these modules in compact form as library files. System programs for building tasks can directly use modules stored in system and user libraries. Two components are provided for this:
LBR is a routine used to create and modify libraries of object modules and Assembler macro directive libraries;
TKB is a task building routine. It builds a task by using files containing object modules and system and user object module libraries. There are three versions of this routine that differ by size of main storage needed, operating time and overlap capabilities.

The system has three components for correcting and debugging:
PAT is a routine used to make corrections in object modules without having to translate the entire module;
ZAP is a routine for making corrections to a file in the form of a task without having to rebuild it. One can also make single corrections to each file;
ODT is a routine to check the process of implementation of a debugged task. It is possible to start the task by parts, modify main storage cells, observe the change in contents of selected storage cells and inhibit a task.

The expanded system of files requires facilities offering the user the capability of managing this system. These facilities include the PIP routine that enables performing the following operations with files: creation, duplication, deletion, expansion, change of name and method of protection, deletion from a catalog and listing of catalogs. The system also has these file management routines:
FLX. This routine permits transferring files from foreign volumes that have a file structure different from that of DOS RT;
VFY. This routine verifies the correctness of file structure on a given volume. One can also make certain corrections to this structure which permits detecting and often eliminating the consequences of possible hardware errors.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



Other DOS RT capabilities include: batched mode of operation, logging of errors and testing of hardware. The first of these capabilities is supported by standard system facilities.

System generation produces a working version of the operating system that is adapted to the specific requirements and hardware configuration. The generation source is the generation data base placed on three cartridge disks with a capacity of 2.5M bytes (for standard distribution of system generator). This base contains a certain basic operating system ready for operation that provides the capability of creating a working system from the working files existing in the base. Generation consists of three phases, each of which is an implementation of a certain set of operator instructions. This simplifies the generation process and the operator need only respond in turn to questions asked about logic, character and numeric values.

DOS RT is designed for minicomputer systems. Because of the efficient control program, expanded application programs and system of files, from a terminal operator's viewpoint, it has many properties of an expanded operating system on a large computer. The system is noted for the ease and efficiency of communication with the user. The system's strongpoint is the capability of convenient use of it in real time when operating with tasks written in both macro assembler and FORTRAN.

COPYRIGHT: Izdatel'stvo "Finansy i statistika", 1981

8545
CSO: 8144/0138



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



SOFTWARE FOR COMMUNICATION BETWEEN COMPUTER SYSTEM AND SPECIALIZED PROCESSORS FOR ARRAY PROCESSING

Moscow VYCHISLITEL'NAYA TEKHNIKA SOTSIALISTICHESKIKH STRAN in Russian No 9, 1981 (signed to press 21 May 81) pp 100-104

[Article by Y. Ivanova and T. Atanasov, scientific associates, People's Republic of Bulgaria]

[Text] Multiprocessor systems offer great possibilities for increasing the throughput of electronic computers. In our opinion, there are two main directions in developing mulitprocessor systems. First is the development of systems of identical processors linked in an appropriate manner. The speed of such systems reaches hundreds of millions of operations per second. However, these systems as a rule have their maximum efficiency when solving problems suitable for parallel computation.

The second is the development of multiprocessor systems from general-purpose and specialized processors. This approach, perhaps, does not yield the gain in throughput as in the first case, but is has a number of substantial advantages: it provides universality of electronic computer systems, offers the capability of increasing throughput as needed, and permits specialization of computer systems in one or more fields of application, etc.

The spectrum of specialized processors developed and under development includes communication and array processors, data base processors, IO processors, processors for processing symbolic information, etc.

Specialized processors for matrix computations are used to process numeric data arrays, large in volume, that occur in fields such as signal analysis, matrix algebra, mathematical statistics, numeric solving of differential equations, etc. Also, in a number of applied fields (geologic and meteorologic research, seismology, military affairs, etc.), methods are used that are characteristic for the above mentioned fields.

Specialized processors for matrix computations are usually connected as peripherals to a computer system, implementing thereby their basic function: freeing the CPU from the large volume of computations involving data matrices. By using array processors, computer system efficiency is raised over ten fold, which depends on the nature of applied tasks being solved, on the ratio between the volume of array processing and data processing in the CPU, on the specific technical parameters of the



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



computer system, etc. An array processor can be made accessible to users through existing programming languages; to do this, software has to be developed to provide communication between a user program and array processor. However, compilers have been developed for the majority of programming languages in use, and including special statements to communicate with an array processor would lead to making changes to compilers too. In connection with this, the array processor access method (MDMP) [APAM] or the main control part of it is built as a subroutine and is activated by a user program by using the standard CALL statement. In the general case, the CALL statement for matrix operations has the following parameters:
mnemonic code of matrix operation;
control information (gives instructions to the APAM on the parameter list structure, on the method for building and executing the channel program, on processing of certain interruptions, etc.);
usually specified for each array (operand or result) are the starting address in main storage (OP) or in the array processor storage, if available; number of elements in the operation; the step defining the relative position of the elements in the operation in main storage; the data format defining the representation of numbers in the array and certain requirements for controlling the sign or modification of the parameters listed above.

When the user requires performance of certain control operations by the APAM or operations that monitor the operation of the array processor, the CALL statement contains parameters differing from those described above.

The main functions of the APAM are:
analysis of CALL parameters and informing the user of any errors made;
generation of the channel program that implements the request or logically complete series of requests for matrix operations;
control of the array processor as a system resource and dispatching of users awaiting this resource;
monitoring the execution of the channel program for processing of inetrruptions or activating user programs used for this purpose.

To perform these functions, the APAM communicates with the user program and the OS control programs. The APAM makes use of certain system blocks and tables; therefore, in constructing it, standard conevtnions for access to system information must be taken into account. Defining the status of the APAM as part of the basic system software depends on a number of factors, for instance on the capabilities of the specific operating system, the size of the APAM, the specific functions that the APAM must implement, etc. Described below are two types of array processors; fundamentally different approaches shaped by the factors listed are used in building methods of access to them.

YeS-2335 Array Processor Software. The YeS-2335 array processor is connected to a system through a channel adapter and is controlled as an ordinary selector channel. The YeS-2335 has direct access to main storage for fetching source data and storing matrix operation results. All arithmetic operations in the YeS-2335 are performed on floating-point data.

The APAM and YeS-2335 provide for execution of the folowing oeprations: vector move and conversion, vector move and conversion of floating-point data to fixed-point, scalar multiplication, vector element-by-element multiplication, inner product of



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



vectors, partial multiplication of a vector, convolution of type of multiplication, sum of vector elements, sum of squares, vector element-by-element addition, squaring of vector elements with sign preservation, and the difference equation.

The APAM also provides for execution of special-purpose operations:
control operations ("no" operation, branch to channel program);
operations for synchronization of user and channel programs (building of channel program, activation of channel program, waiting for end of channel program);
operations for specifying addresses of user subroutines for interruption processing (determination of subroutine for processing interruptions during YeS-2335 operation, determination of subroutine for processing program controlled interruptions);
operation for determining the mode of syntactic analysis performed by the APAM.

The APAM is an Assembler program. Using conventional assembly, it can be generated in different versions as a function of user requirements and specific operating system characteristics. Since it is not very big, it is entirely connected to the user program written in PL/1, FORTRAN or Assembler, and functions as an ordinary subroutine. The YeS-2335 APAM does not require the user to have a detailed knowledge of systems programming. For each user request defined by an appropriate CALL statement, it builds a channel program and a series of descriptors of arrays-operands and matrix-result. After execution of the channel program has started, the user may obtain control for completing processing in the CPU while the array processor is executing its tasks concurrently with it.

The diagram shown in fig. 1 gives a most general idea of the method by which a request for a matrix operation is satisfied in the YeS-2335.

Software for Array Processor with Its Own Storage. An array processor with its own storage is connected to a computer system as a peripheral through the block-multiplexer channel. The set of operations performed by it is considerably larger than the YeS-2335 set. Operations are provided for regular and inverse Fourier transforms, square interpolation, Weiner-Levinson filter, complex multiplications, etc.

The array processor has up to 1M bytes of its own storage. A user can divide it into a maximum of seven partitions. An independent task can be activated in each partition. A user has up to 15 registers for each partition. A rather complex algorithm can be implemented to solve a problem by making use of the registers and index arithmetic facilities (22 operations).

The APAM for an array processor with its own storage is a large, structurally complex program that is a subsystem of the operating system. It is activated by a standard CALL statement in the user program written in PL/1, FORTRAN or Assembler, but only its control part is tied to a user program as a subroutine.

The APAM has the following main components:
monitor that maintains communication with the console operator all allows execution of commands from the console to manage the resources of the array processor;
control programs that define and execute user tasks in the array processor;
EXIT programs that support and control the array processor with its own storage at the system level.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



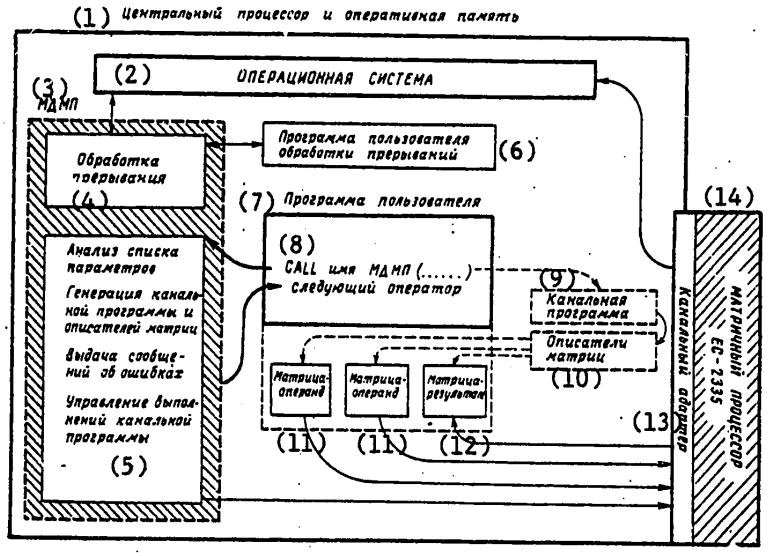



Fig. 1. Functions of the Array Processor Access Method for the YeS-2335 Array Processor

Key:

1. CPU and main storage
2. operating system
3. array processor access method
4. interruption processing
5. parameter list analysis, channel program and array discriptors generation, issuing of error messages, channel program execution control
6. user program for interrupt processing
7. user program
8. CALL name APAM (.....) next statement
9. channel program
10. descriptors of arrays
11. array-operand
12. array-result
13. channel adapter
14. YeS-2335 array processor

By using a series of CALL statements, a user program calls to the APAM (with the name XXXX), and through special statements, the user marks the beginning and end of a task (fig. 2). The APAM defines the task, building a channel program (KP) and control table (UT) that contains the task coded by a special method. During execution of the channel program, the control tables and arrays-operands (X and U) are moved into array processor storage. And finally, after execution of the algorithm specified by the control tables, the array-result (Y) is moved into main storage. The availability of its own storage in the array processor and the capability of parallel execution of tasks in its partitions provides the capability of making the maximum efficient use of a computer system when suitable planning of the operation is performed.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



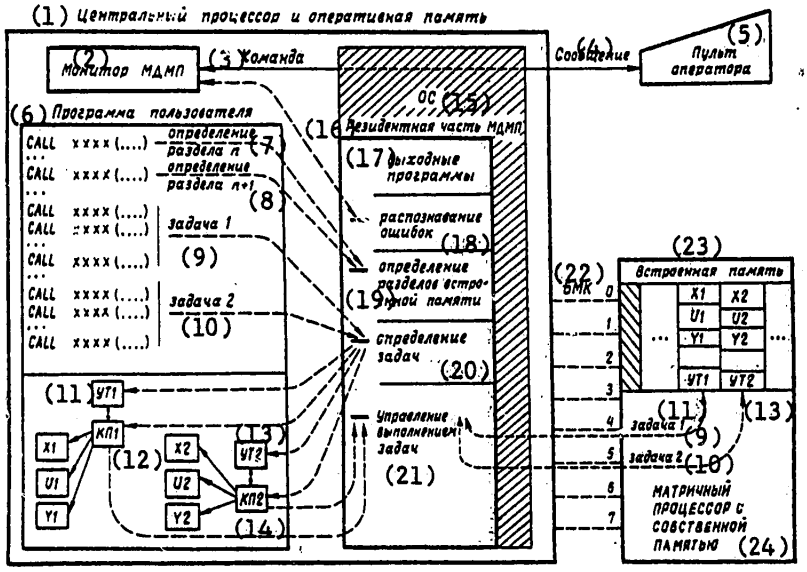



Fig. 2. Structure of the Array Processor Access Method for an Array Processor with Its Own Storage

Key:

1. CPU and main storage
2. APAM monitor
3. command
4. message
5. operator console
6. user program
7. definition of partition n
8. definition of partition n+1
9. task 1
10. task 2
11. UT1 = control table 1
12. KP1 = channel program 1
13. UT2 = control table 2
14. KP2 = channel program 2
15. OS
16. resident part of APAM
17. exit programs
18. recognition of errors
19. definition of partitions of built-in storage
20. definition of tasks
21. control of execution of tasks
22. BMK = block-multiplexer channel
23. built-in storage
24. array processor with its own storage

COPYRIGHT: Izdatel'stvo "Finansy i statistika", 1981

8545
CSO: 8144/0138



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



AUTOMATED SYSTEM FOR MASS PREVENTIVE TREATMENT OF POPULATION (SAMPO)

Moscow VYCHISLITEL'NAYA TEKHNIKA SOTSIALISTICHESKIKH STRAN in Russian No 9, 1981 (signed to press 21 May 81) pp 127-133

[Article by G. Astardzhiyan, candidate of engineering science, B. Bayev, scientific associate, and B. Raychev, candidate of engineering science, all from People's Republic of Bulgaria]

[Text] Mass preventive treatment of the population is an exceptionally importa part of the work of public health services. It is noted for high labor-intensi - ness and involves processing a large amount of varied data. In connection with this, there is the question of automating treatment by using modern computer technology. To do this, it is necessary to equip departments of mass preventive treatment with terminals, teach medical personnel to work with an automated system, use some of the computer resources of a given center, and make available specialized software that operates in real time.

The automated system for amss preventive treatment, SAMPO, uses an applications program package (PPP) designed to operate in real time under control of the SUIP PPP together with YeSTEL and YeS EVM hardware.

The SUIP package, operating in real time, supports a terminal network built on the basis of YeSTEL hardware and a centralized data base. SUIP offers facilities for interface (macro instructions), with the use of which user programs make use of the capabilities of this package. SUIP operates in one partition (with an optional high priority) of main storage, and all application programs operate under its control in this same partition. As a result, only part of the computer resources in a center are used by the system. The other partitions under multiprogramming operation may be used for batched processing.

The hardware supported by the SUIP PPP includes:
the YeS-8501 electromechanical terminal or the IZOT 8531-Ye1 on switched or dedicated communication channels;
the VTS 56100 video terminal or IZOT 7925 together with a YeS-7186 or YeS-7187 printer on dedicated channels;
the IZOT 8500 miniterminal on dedicated channels. This terminal is convenient for entering numeric data during laboratory tests;
telegraph apparatus on switched or dedicated channel;
a pseudoterminal consisting of the YeS-6012 card reader and the YeS-7033 alphanumeric printer.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



The SAMPO package can operate with the YeS-1020 type computer. Sufficient for it is the hardware in the standard configuration for this computer and the hardware in the YeSTEL teleprocessing system, including the YeS-8401 multiplexer and the required line apparatus to link the remote terminals mentioned above.

The SAMPO PPP consist of one program operating in the batched processing mode (autonomously of SUIP) and eight programs operating in the real time mode under control of the SUIP telecommunications monitor. The following program functions are provided in the real time mode:
a) logging and issuing a number for a patient that has arrived (registration function). Registration includes:
obtaining a free record from the main file for the new patient to be treated;
entry of last name, first name and patronymic for patient to be treated;
definition of a personal number for the patient to be treated. The personal number is defined by the system and is used to identify the patient;
b) compilation of information on the number and names of people registered in the system (information function). Three types of queries are possible:
query on the number of registered patients;
query on names and SAMPO numbers of registered patients;
query on names and SAMPO numbers of registered patients with specific initials;
c) system-terminal operator interaction during which various patient data are entered on disk storage (conversation function). Data are entered under control of a special questionnaire prepared in advance by a systems programmer. During the conversation, the operator responds to questions, entering a specified set of data (text and numeric requisites, coded values). When a question on a numeric requisite is asked, the system reports the limits in which a response is valid. A special category of requisites are the so-called coded requisites, each of which may assume one or more text values specified in advance. All valid texts are stored in a SAMPO text file that is a system thesaurus. Each text is coded with the aid of a specified number called the elementary coded value (EKZ) [ECV]. Elementary coded values may be only numbers that meet the following formal requirement: only one decimal position of the ECV can differ from zero. For example, 0, 1, 2, 3, 10, 20, 30, 100, 1000 may be ECV's, but 11, 33, 331, 157 and 202 may not.

Numbers that cannot be ECV's (i.e. numbers having more than one nonzero decimal position) are considered as compositions of ECV's. For example, 11 is considered a composition of the ECV's 10 and 1. Each requisite corresponds to one set of ECV's that are called proper ECV's. Each proper ECV corresponds to a specific text from the system thesaurus. A coded requisite can assume only those values that are proper ECV's or are compositions of its proper ECV's. Valid values of coded requisites are determined again by using validity limits;

d) entry of a value of a specific requisite determined by a terminal operator (entry function). When the indicated requisite has already been assigned a value, the system issues a warning message and displays the value (if there is no protection from unauthorized reading) before asking a question on the new value. This program function in many respects is similar to the preceding;

e) compilation of automatic diagnostic evaluation of patient status. This program function is declared by including the suitable element in the appropriate questionnaire (AMPQ TYPE-SET). It is used to analyze the values of a set of digital requisites (requisites-arguments) and select one requisite, called the requisite-function. The requisite-function must have the functional designation "diagnosis SAMPO," be



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



coded and have only two valid values: 1 means the patient has the corresponding illness, and 0 means he does not.

f) compilation of patient status report (report function). This program function is used by the operator to request the system compile a patient status report. In requesting the report, the operator can define what requisites (with assigned value, without an assigned value, outside normal limits, with normal values) and what groups of requisites are to be included in the report. The following groups of requisites can be named: identification, anamnesis, interpretation, laboratory, physiological tests, SAMPO diagnosis, physician's diagnosis. The report can be sent to a specific terminal or the name of the external subscriber PRINT can be used (which is most often a alphanumeric printer);

g) deletion of patient data and storage of it in an archive file on magnetic tape. This program function deletes a record in the main file, keeping in the process the accumulated data, entirely for the TAPE function and partially for the TRAC function.

Besides these basic functions, the SAMPO system performs certain special functions: it uses SUIP facilities to limit access to certain program functions, provides protection for reading and writing of individual requisites and questionnaires (used by the conversation function), maintains identification of the originator of certain requisites (stores the initials of the terminal operator that entered these requisites), and uses certain internal SUIP mechanisms, for example automatic start of a function and automatic output of data accumulated to a given terminal.

Fig. 1 is a simplified diagram of the operating environment of the SAMPO PPP. The data base of the SAMPO package determines the structure and composition of data needed to execute the various program functions in the system and also their internal organization. The data base includes files (data sets) needed for SAMPO package functioning and facilities to describe the makeup and structure of data in the files (tables).

SAMPO data files are part of the overall system of files supported by the SUIP package. Two SAMPO files are included in the SUIP file management table: the main file, MAINFIL, and the text file, TEXTFIL.

The SAMPO systems programmer has available language facilities (macro instructions) to generate the so-called distributing table that describes the makeup and physical structure of data in the main file. The distributing table is defined in SUIP as an applications program stored in the real-time program library. When needed, it is loaded into main storage by using the SUIP macro instruction FQWPC TYPE-LOAD.

Each record in the main file (except a "zero record") can be used by the system to keep data pertaining to a treated patient. Records are formed and released dynamically. The zero record contains information as to which records are being used and which are free. The main file has the direct organization with unblocked fixed-length records, which is selected by the SAMPO systems programmer.

Ordinary records in the main file include fields that are the individual requisites.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



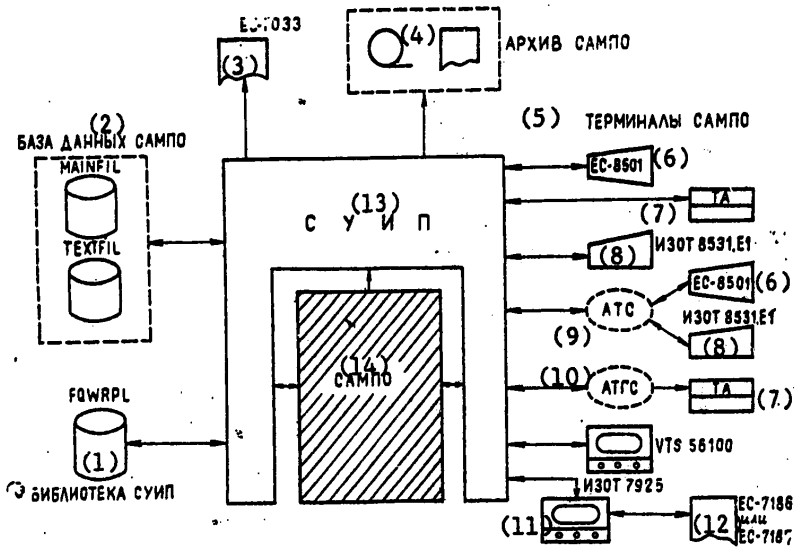



Fig. 1. SAMPO package operating environment

Key:

1. SUIP library
2. SAMPO data base
3. YeS-7033
4. SAMPO archives
5. SAMPO terminals
6. YeS-8501
7. TA [telegraph apparatus]
8. IZOT 8531.Yel
9. ATS [expansion unknown]
10. ATGS [expansion unknown]
11. IZOT 7925
12. YeS-7186 or YeS-7187
13. SUIP
14. SAMPO

Fig. 2 shows the format, structure and organization of record N, where N = 1, 2, ..., the number of records (of patients) in the main file, and also the requisite field $R_X$, where X = 1, 2, ..., M, and M is the number of requisites. The record must be long enough to hold all the requisites defined in the distributing table for a main file record.

The distributing table (fig. 3) contains information on the initial length, type, function and other such attributes of each field in main file records. This table is compiled by using a special macro instruction, and is assembled and placed in the real time SUIP library. Each filed in a main file record is described by using an eight-byte element, the structure of which is shown in fig. 3.

The text file is the general system thesaurus. It holds the designations of all requisites and elementary coded values (ECV's) and the limits for valid and normal digital requisites.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



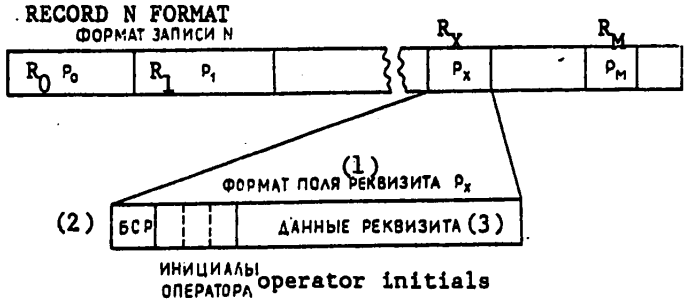



Fig. 2. Structure of main file record

Key:

1. requisite field $R_X$ format
2. BSR [expansion unknown]
3. requisite data

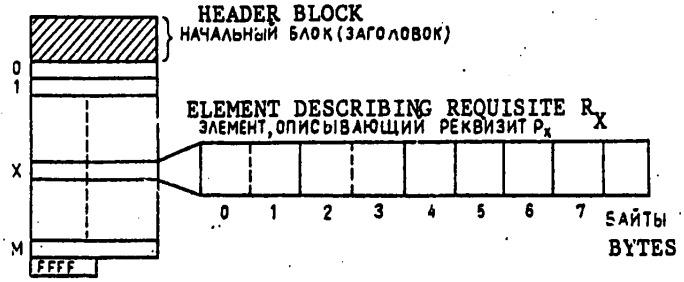



Fig. 3. Structure of distributing table

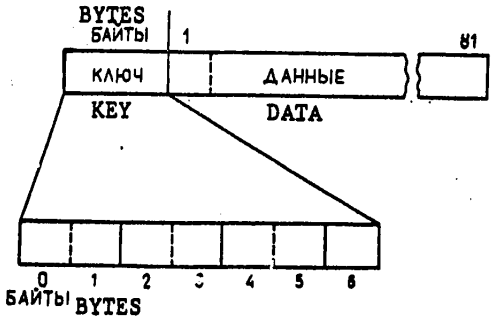



Fig. 4. Structure of logical record in text file



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



All information in the text file can be repeatedly duplicated by using various "styles." Here a style is possible mode of operation with the SAMPO system, i.e operation of the system simultaneously in several languages, use of various forms for asking questions, etc. A style identifier has two bytes.

The text file has the index-sequential organization. Records are blocked and fixed-length. A logical record has 88 bytes: 7 bytes for the key and 81 bytes for data. The key takes up the first seven bytes of the logical record. The structure of the key and data in the text file logical record is shown in fig. 4.

The text file may contain records of various styles. For example, one style may be used for asking questions and another for compiling a report. Various styles can also be used when the system must operate in several languages. The main (standard) style is style A.

The text file is created in the batched processing mode (autonomously of SUIP) by using a special program. Two input cards are required for each text file record: a card describing the record key, and a card containing teh data.

The punched card describing the key has special syntax and must precede the punched card with the data.

SAMPO includes facilities operating in real time for making corrections to the text file. Used for this are four transactions that implement the following functions: add a new record to the text file, delete a record, replace a record and change a record.

All the SAMPO functions discussed have been implemented as SUIP transactions. Thus, the SAMPO operator uses a relatively elementary and convenient command language to make requests to the system. Also, the SAMPO systems programmer has program facilities. Thus, the data base offers great conveniences in implementing all program functions and expanding the SAMPO system.

The data base is noted for its universality and flexible structure. Formal replacement of the contents of data in the data base (files and dsitributing table) and keeping the same structure permits using it not only in public health, but also in other fields. The software package is delivered in source code; therefore, the programs can be easily modified.

COPYRIGHT: Izdatel'stvo "Finansy i statistika", 1981

8545
CSO: 8144/0138



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



DATA ACQUISITION AND PROCESSING SYSTEM FOR USSR GAS INDUSTRY

Moscow VYCHISLITEL'NAYA TEKHNIKA SOTSIALISTICHESKIKH STRAN in Russian No 9, 1981 (signed to press 21 May 81) pp 133-138

[Article by N. Moletz, engineer, Hungarian People's Republic]

[Text] Close cooperation in developing software for the YeS-1010 computer, especially in developing and improving application software packages, has been established between the Videoton plant and the Institute for Organization and Computer Technology in Metallurgy and Machine Building. A specific achievement from this cooperation is the information system for the USSR Ministry of the Gas Industry that provides on-line and statistical information on the production, use and export-import of gas. Technical requirements for the system were defined by specialists in the USSR Ministry of the Gas Industry (Mingazprom).

Task Description. The system for acquisition and querying of gas information has two functional parts.

The dispatch part provides for acquisition, storage and processing of on-line and daily production data, and for presenting entry and calculated data on displays and printers. This information is used primarily to prepare gas production and distribution reports. Data is acquired daily and repeatedly (a maximum of 12 times a day at this stage) from remote terminals operating synchronously with a central computer.

The most important dispatch data are the pressure at certain points in the pipeline, temperature and flow rate of the gas, data on conditions of the compressor stations, and periodic production indicators from gas production sites.

The commercial part provides data on gas production, distribution and use for relatively long periods (a day, 10 days, a month, quarter, half-year and a year). This information makes it possible to evaluate the operation of the gas industry and compare plan data to actual. Thus, the system promotes raising development according to plan of gas production and distribution.

System Requirements. In the statement of the problem, the Soviet specialists described in detail the content of the system input messages, queries, and principles of operation. The system has to support reception of data arriving from 10 remote information stations, periodic processing of it (every hour), continual capability of querying data on the 12 displays or printers connected to these displays, and continuing management of the daily log of system operation.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



Messages on the display screens are called "standard items." This means that the system must display on the screen the data according to previously specified algorithms. The display operator does not have the capability of specifying a new algorithm. He can only select from those existing and list, restart or terminate data queries.

Hardware. Hardware requirements were based on two YeS-1010 computers (fig. 1) with a storage capacity of 64K bytes each and with an identical set of peripherals: one fixed-head magentic disk storage unit, two replaceable magentic disk storage units, one printer, a paper tape station and a general-purpose interface for data transfer.

In the initial state, 10 synchronous terminals are connected to one computer through the interface, and 12 asynchronous terminals are connected by modems to the other one. The capability for expanding the set of terminals had to be provided in developing the software system.

Principles of Solution. Designing the software system involved solving two groups of problems. The requirements on operation and reliability are important for system functioning. Use of the system oeprating in real time is largely a function of response time, i.e. the time that elapses from data query time to answer time.

Storage of a large amount of data is possible only on packs of disks with movable heads, the access time of which is relatively long. Design of data placement largely depends on the amount of data. When the entire system is fully used, the average amount of data collected per day during the 12 polling cycles is about 2M bytes. Synchronous lines for data transmission under noise conditions can operate at the rate of 600 bits/sec. Theoretically, this requires 4 minutes for each line in each cycle. However, the work done by an operator on a terminal also has to be considered (for example, replacement of reels, paper tape, etc.). Therefore, the full time of one reception is about 10 to 15 minutes.

Because the processing programs can service several files at the same time, their operation can be considerably speeded up by placing the data to be processed and that already processed on separate disk units. In this case, the time for searching on disks and thereby the program run time can be reduced. The expected run time, considering the data base sizes (defined as about 5M bytes), is about 15 to 20 minutes. During processing, the data base contains partially updated data, but the capability of querying it is provided.

There are two forms of data representation in the system. With the first, data does not need conversion and response tiemis about 1 second. With the second, data can be put out to the screen only after conversion. It is very important that the data conversion time be short. It is advisable to use separate storage units to hold the data base and the file of responses being generated. Therefore, the contents of two disk units are divided. The BASIS pack holds the data files, and the TRANSIT pack holds the input messages and responses formed earlier and during queries. As a function of the contents and amount of data, the time for forming responses is 5 to 15 seconds.

The second, but no less important, group of problems is the development of software and hardware solutions that promote the flexibility of programming and reducing program debugging time. The problem is solved on the basis of a monitor for multitask



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



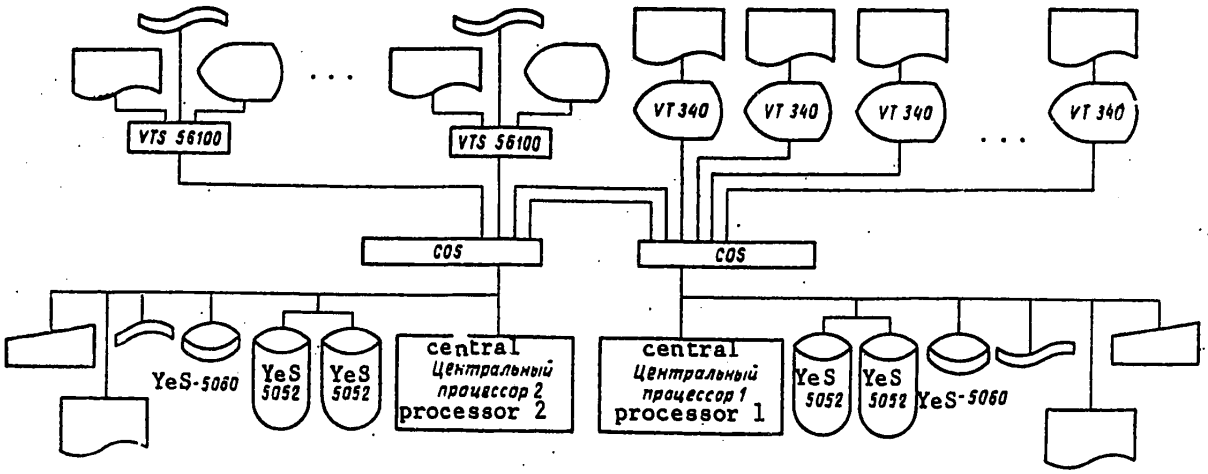



Fig. 1. Composition of hardware for the USSR gas industry information system

processing which enables solving simultaneously up to 96 user tasks. For this it is sufficient that the displays and synchronous terminals be served by independent programs, and that complex calculations can be divided into partial tasks and that final results of the calculations can be derived by linking these partial tasks. Because of this system structure, for example, there is no problem in controlling the programs that service the several displays and synchronous terminals, because this is done by the monitor. By considering certain general instructions for programmers, programs can be written and debugged individually, as if the system had no other elements. Flexibility of programs is enabled by the macro assembler.

Its value is that definitions common to the various programs (for example, structure of files, distribution of tasks, etc.) are defined in one place; thus, changes needed for programs already prepared require only retranslation of the source text. Standard modules (for example error messages) have been created to facilitate the work of the programmers.

Task Solution. From the viewpoint of functions performed, the gas industry information system can be divided into three subsystems: acquisition, processing and querying of data. The same program is run on both computers. The composition of terminals can be identical, thanks to which the number of terminals in use can be doubled. Fig. 2 shows a diagram of the operation of the programs, and the data stream of one computer.

The data acquisition subsystem transfers data between the synchronous terminals and the TRANSIT magnetic disk. Data acquisition may be initiated by a synchronous terminal operator. Message entry is performed by the program serving the synchronous line. This program stores the message on a minidisk, which in case of erroneous or damaged messages does not require deletion of them. At the same time, because of its rapid operation, messages arriving simultaneously can be received.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



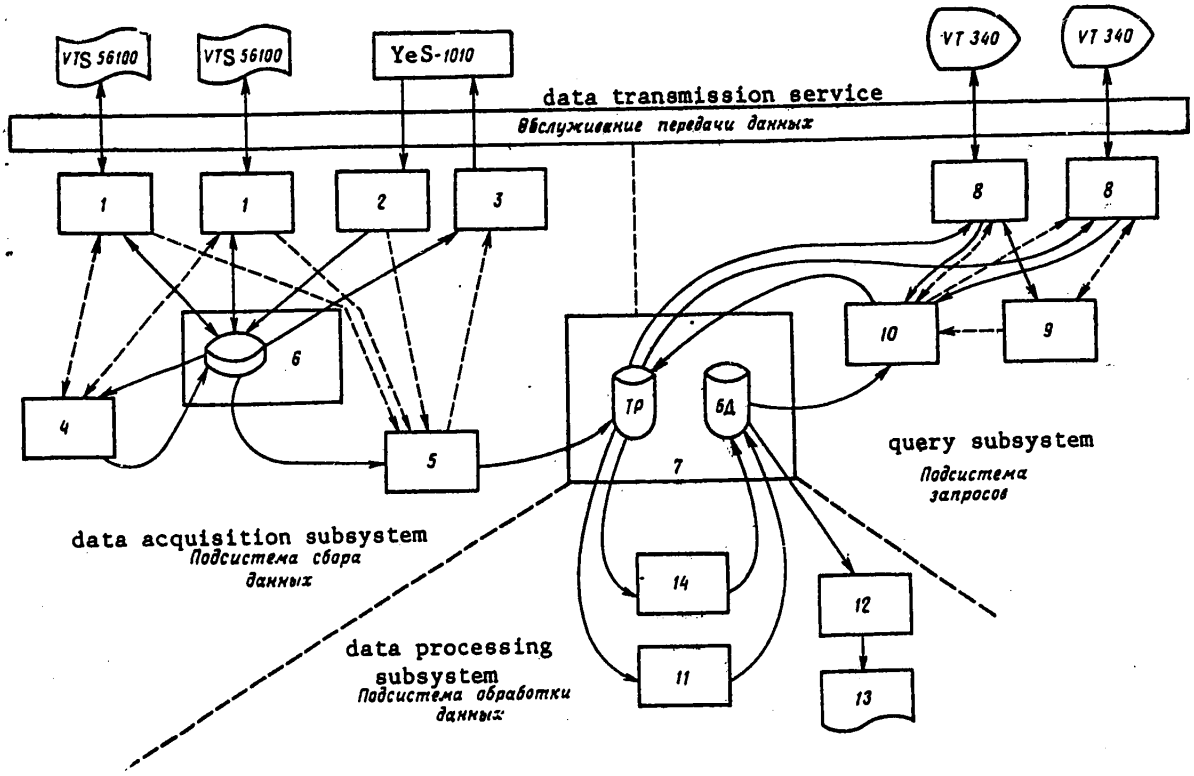



Fig. 2. Structure of information system for Ministry of Gas Industry

1. service of synchronous lines
2. entry into YeS-1010
3. YeS-1010 output
4. message analyzer
5. archives
6. maintenance of files on fixed-head disk
7. maintenance of files on removable disk [left, TRANSIT pack; right BASIS pack, data base]
8. service of asynchronous lines
9. command analyzer
10. response generator
11. commercial processing
12. programs for printing lists
13. printer
14. dispatch processing

← data flow

<-- - - logical link between programs

The entered complete message is repeated syntactically by one task common to the synchronous terminals. In event of an error, the message read together with notes of errors is returned by the line operator. The latter sends this message back to the data source. When there is no error, he provides for message storage on the TRANSIT disk; in the process, the line operator can receive a new message. After placing the message for storage, the operator sends the message to the TRANSIT disk of the other computer through the "computer-computer" task. The message, arriving from one computer to the other, is checked syntactically; therefore, it can be sent for storage right away.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



Programs in the processing subsystem are started directly by the central computer operator at the end of the data acquisition period. While processing messages coming in, the programs update the data base and prepare the data files for query.

The query subsystem is always ready for work; it is activated by dispatchers from the asynchronous terminals by using commands that call data to the displays. The command read is checked syntactically by the system. Any error message is sent back to the display. When no errors are detected, the response "Ready" is sent immediately and after the program run, a response is generated and displayed on the screen. After displaying the response, the terminal awaits a new query.

Conclusion. The software system developed meets the parameters specified earlier. Expansion is provided for. All capabilities of the monitor (96 tasks) are not yet exhausted. After expanding the system with terminals (to 18 synchronous and 24 asynchronous), another 32 programs can be included in the system.

For system functioning, it is disadvantageous to have all the hardware being fully utilized, since if a disk unit goes down, the given system configuration could operate only by reducing the number of functions performed. Joint operation of two computers is possible also in this case.

The software system developed has special and common elements that are easily separated from each other. After replacing special with corresponding common elements, the system can be used, for example, to service rail ticket booths, hotel reservations, to monitor production processes and in other areas where rapid information service is needed.

COPYRIGHT: Izdatel'stvo "Finansy i statistika", 1981

8545
CSO: 8144/0138



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



ACQUISITION OF DATA ON COMPUTER HARDWARE OPERATION IN GERMAN DEMOCRATIC REPUBLIC

Moscow VYCHISLITEL'NAYA TEKHNIKA SOTSIALISTICHESKIKH STRAN in Russian No 9, 1981 (signed to press 21 May 81) pp 149-154

[Article by E. Hanisch, engineer, German Democratic Republic]

[Text] In ensuring product quality, a large role is played by the evaluation of the entire set of data on the products. However, its is easy to analyze product quality at the development and production stage, but acquisition and evaluation of data on product performance during operation entails considerable difficulties. This is because products at the using site are not always serviced and monitored by the manufacturer. Consequently, the user must take part in acquiring data on product performance during operation. The need to acquire such data is especially important in the case of costly and high-performance computer hardware, since the manufacturer obtains the capability of taking suitable steps and thereby avoiding large losses when there is deviation between actual and planned reliability indicators.

Results of acquisition of data on product performance during operation basically can be used in the following areas:
to define achieved indicators of product reliability during operation to confirm planned reliability of a given product, to derive reliability indicators being planned for new and updated products, to solve general guarantee problems and to confirm or reject user claims during the guarantee period;
to study typical malfunctions to take suitable steps to ensure and enhance product reliability and to improve the design, technology and production of new and updated products;
to plan for the supply of spares;
to evaluate the efficiency and quality of maintenance; and
to compile optimal procedures for maintaining products in good working order.

In connection with such broad capabilities for making use of this data, manufacturers often request a very large amount of information from the user on their products. As a rule, this leads to making excessive demands on the user and as a result, to failure. However, if the amount of data is limited, costs for generalizing the data by the manufacturer are reduced, and with suitable organizational measures, the information feedback system can be adjusted.

An information feedback system used for hardware, computers in particular, has been operating in the German Democratic Republic since 1975. Experience gained in this time confirms this method is justified in practice. Success is due primarily to



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



the following reasons:
maximum consideration for user interests, in particular in the design of forms;
creation of a data collection form that compels the information provider to fill out the form right at the time the event occurs;
use of existing organizational forms;
acquisition of local information only in the amount needed so as to limit acqusition costs;
providing checks to limit the amount of erroneous data.

Computer hardware is very complex; therefore, full collection of precise data is very difficult. It turned out that it is expedient to gather local data on two separate forms, especially with regard to the classification of local data providers (operator, engineer) at a computer center. So we use the computer log form (fig. 1) and the malfunction accounting form (fig. 2) with an attached accompanying card for an assembly when it is replaced. These forms basically meet the recommendations for use within the Unified System of Computers, but in content they were designed specially for use in the GDR. The computer operator fills in the log form. It is used to gather chronologically data on all operations, on all productive and nonproductive periods and on all events that occur. Thus, the data in the log not only defines the reliability indicators, but is used by the user for his own accounting.

Information on failures (analysis, cause, elimination and location of failure) is recorded by engineers on a malfunction accounting form. The data on both forms supplement each other and thus the capability is provided for checking missing data.

In contrast to the usual collection systems, in our system the user compiles a so-called monthly report based on the computer log. Contained in it are information on the installation configuration and generalized information for determining the indicators for the reliability of the installation and the individual hardware. Generalization of the computer log data and compilation of the monthly report have to be done by the user, because it is not possible to generalize and evaluate centrally the entire amount of computer log data generated by all users at the manufacturing plant. This method also has the advantage that incorrect and missing data in the computer log can be corrected and supplemented by the user. In the case of centralized evaluation of local data at the manufacturing plants, correction of the forms is not possible or the possibility of correction entails large costs.

Feedback is organized as follows. At prescribed schedules, the user sends the malfunction accounting forms and the monthly report to the competent agency for maintenance at the manufacturing plant. The maintenance agency checks the collected forms for completeness and accuracy of data and makes corrections when necessary. Also, an evaluation of the data is made with regard to the specific user conditions; when necessary, instructions are sent to maintenance teams on steps to take to ensure reliability. The monthly reports are forwarded to maintenance agencies at a centralized manufacturing center for further reduction of the data and evaluation of it. Management decisions are made according to the results. Then, after suitable preparation, information on operating reliability achieved is sent to management staffs, development plants, unit manufacturers, foreign trade agencies and the Coordination Center for the Intergovernmental Commission on Cooperation among the Socialist Countries in Computer Technology.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6






Fig. 1. Computer log form

Key [left to right, top row]:

computer supplier
computer model number
1. card code
2. computer code
3. plant computer number
4. date form completed
5. sheet
temperature, °C
relative humidity

[left to right, bottom row]

6. user code order number
7. BPO [expansion unknown]
8. job title
9. operation code
10. operation time; from, to
11. total operation time
device address
external indications of malfunction
time of malfunction
12. start
13. end
14. meter reading
specialist call time
15. called
16. arrived
17. operator special comments
18. line number
signature; operator, engineer





APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

FOR OFFICIAL USE ONLY

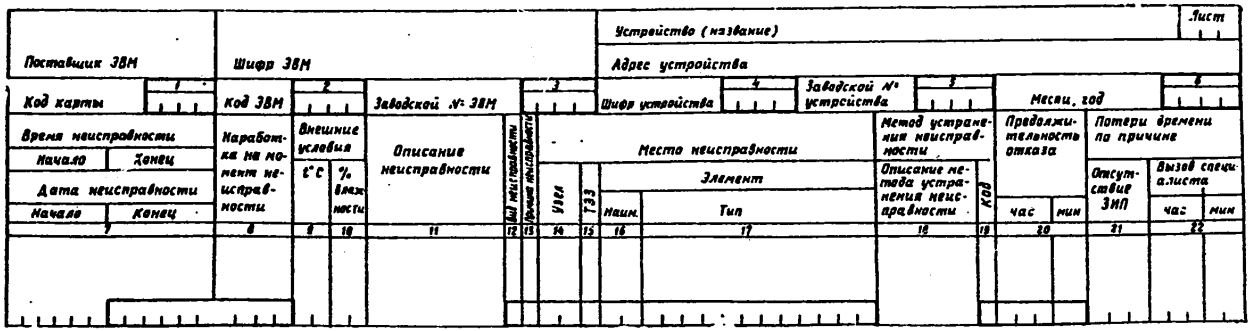



Fig. 2. Malfunction accounting form

Key [left to right, top row]:

computer supplier
computer model number
device (name)
sheet
device address

[left to right, middle and bottom rows]

1. card code
2. computer code
3. plant computer number
4. device model number
5. plant device number
6. month, year
7. time of malfunction; start, end
   date of malfunction; start, end
8. accrued operating time to time of malfunction
   environment
9. temperature, °C
10. % humidity
11. description of malfunction
12. type of malfunction
13. cause of malfunction
    location of malfunction:
14. assembly
15. TEZ [card]
    element:
16. title
17. type
    method of eliminating malfunction:
18. description of malfunction elimination method
19. code
20. downtime; hours, minutes
    time lost due to:
21. lack of ZIP [kit of spare parts, tools and accessories]
22. specialist call; hours, minutes



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

FOR OFFICIAL USE ONLY

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



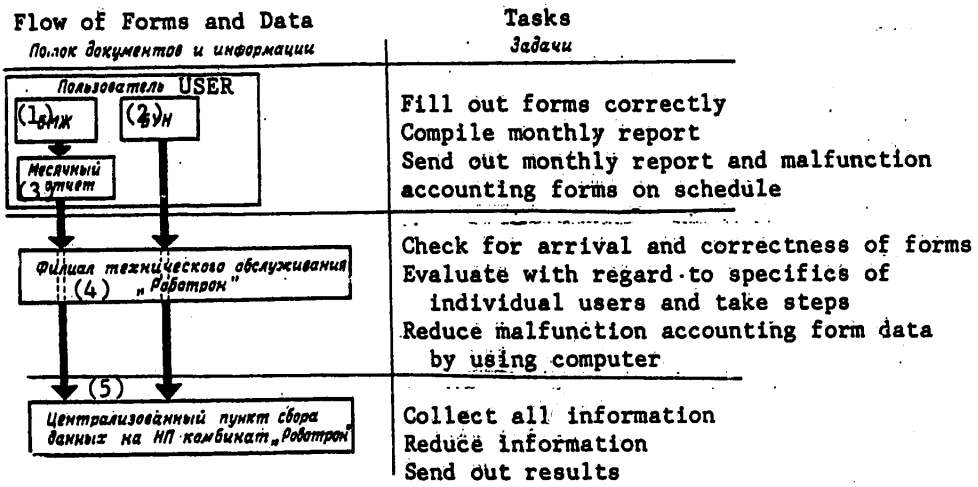



Fig. 3. Organization of system for monitoring product performance and feedback information in the German Democratic Republic

Key:
1. BMZh [computer log form]
2. BUN [malfunction accounting form]
3. monthly report
4. Robotron maintenance branch
5. data collection center at Robotron NP [scientific production] combine

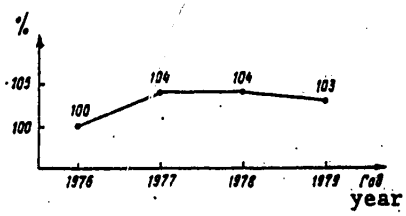



Fig. 4. Increase in YeS-1040 system availability factor

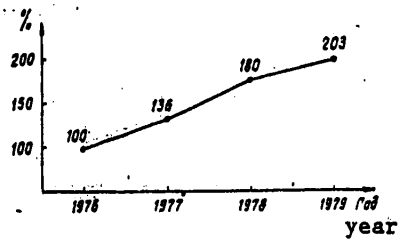



Fig. 5. Increase in mean time to failure of YeS-2640 central processor

Data from the malfunction accounting forms is condensed by using a computer at the manufacturer's data center and printed out. The reports are sent to the device developers and manufacturers and to the maintenance agency for failure cause analysis (fig. 3).

The data in the monthly reports and malfunction accounting forms provide the capability of making numerous evaluations. Let us list just those applied in the GDR: monthly account of reliability indicators achieved for individual installations



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



(with division by users) within the guarantee period to check the indicators stipulated in the warranty;
generalization of monthly reports compiled by users for calculating reliability indicators for installations and individual devices to confirm a specific reliability, and the calculation is made from different points of view, for example for various reporting periods (month, quarter, half-year, year, period with a change in design and technology, etc.), by different installation sites (export computer, domestic computer);
consideration of the main direction in the development of efforts to improve quality;
analysis of the causes of failures of the same type of devices.

The method, used in the GDR in recent years, to collect data on the operation of devices has become a tool to exert a continual effect on improving product quality and reliability. In contracts made, the user assumes the obligation of taking part in this effort, which is a prerequisite for maintaining the warranty.

In the GDR, all Unified System computers are subject to continual checking within the warranty period, and sometimes even after it expires. Information is gathered on the operating performance of all export computers, in particular those from the USSR, in accordance with contract conditions.

Since the day the system was introduced, much practical data has been gathered on product operating performance. As a result, the capability has emerged of concretely determining the development of efforts on improving product quality and of evaluating the effectiveness of steps taken to raise quality. The effect of the feedback system fundamentally depends on the type, quantity and use of results for solving theoretical problems of reliability and for commercial purposes, etc., and a considerable effect is due to the continuous monitoring of the maintenance of the planned reliability. Comparison of indicators achieved with those planned makes it possible to timely and directly affect the quality and reliability of installations where planned indicators have not been reached. The effectiveness of all steps taken ensures a continual increase in product quality and stability of reliability. The data on the YeS-1040 installation availability factor and the YeS-2640 CPU mean time to failure (figs. 4, 5) can be cited as an example.

COPYRIGHT: Izdatel'stvo "Finansy i statistika", 1981

8545
CSO: 8144/0138



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



## NEW SMALL COMPUTER HARDWARE FROM GERMAN DEMOCRATIC REPUBLIC

Moscow VYCHISLITEL'NAYA TEKHNIKA SOTSIALISTICHESKIKH STRAN in Russian No 9, 1981 (signed to press 21 May 81) pp 155-161

[Article by K. Leidler, engineer, German Democratic Republic]

[Text] The SM-1624 microcomputer (Robotron K-1510) is an eight-bit, low-throughput computer for handling computational, testing and control tasks. It features: design and functional modularity and flexibility, i.e. the capability of being used within other equipment, low power consumption, high CPU reliability and small dimensions.

Because of these properties, the computer is used primarily as a module in automatic devices (measuring, regulating, control and testing), laboratory devices in test beds, and in teleprocessing devices.

The element base in the microprocessor are LSI circuitry for the arithmetic and logic unit with PMOS technology and additional TTL elements with medium scale integration.

Central processor specifications:

| | |
|---|---|
| width, bits | 8 |
| instruction execution time, microsec. | 12.5 . . . 46.0 (20,000 operations/sec.) |
| processor cycle, microsec. | 4.17 |
| data format, byte | 1 |
| instruction format, bytes | 1, 2, 3 |
| number of basic instructions | 48 |
| size of addressable area, K bytes | 16K |
| addressing | direct logging, direct storage, indirect storage, immediate |
| I/O address area | 16 addresses for input area, 48 for output |
| interrupt system | one interrupt level, marked; eight programmable interrupt sublevels |
| storage: main | 1K bytes, LSI |
| ROM | 2K bytes, LSI |

The battery module permits retention of data in main storage when power is interrupted.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



The SM-1624 computer common bus is divided structurally into long and short. Connected to the short bus are the CPU, supplement to the CPU and devices for interface to the control console, and the PROM programming unit. To the long bus, one can arbitrarily connect storage TEZ [cards] and interface units that implement standard or special interfaces.

One can also connect a clock and I/O units for digital data for 16 signals.

The SM-1624 microcomputer is a modular unit with a width of 440 mm for mounting in racks meeting the recommendations of the MKYe [expansion unknown].

The module holds a maximum of twenty 135 x 170 mm cards. The microcomputer bus is made in the form of a printed circuit board on a response panel.

The computer software has several components. To prepare programs, there is the CROSS assembler for the Robotron 4000 family of small computers. There are also system programs to diagnose errors, programs for programming the PROM and utilities for preparing special programs in special operating cases.

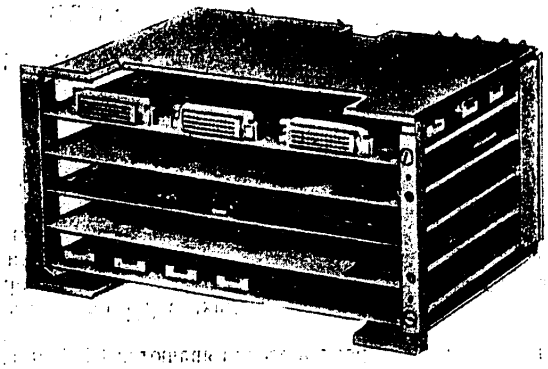


Fig. 1. SM-1626 microcomputer

The SM-1626 microcomputer (Robotron K-1520) is an eight-bit microcomputer (fig. 1) with a higher scale of integration and higher throughput than the SM-1624 model. The SM-1626 microcomputer is used for automating control of production, automation of tests, in laboratory complexes in systems for processing information for scientific and technical and economic tasks of control and checking.

Cards may be in any place in the central processor bus. For small tasks, one card with the central processor without a computer bus is sufficient. The element base for the microprocessor is LSI circuitry for the arithmetic and logic unit with NMOS technology.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



CPU specifications:

| | |
|---|---|
| width, bits | 8 |
| instruction execution time, microsec. | 1.6 . . . 9.2 |
| system step, ns | 407 |
| data format, bytes | 1, 2 |
| instruction format, bytes | 1, 2, 3, 4 |
| number of basic instructions | 158 |
| size of addressable area, bytes | 64K (can be expanded to 128K bytes) |
| I/O address area | 256 input addresses; 256 output addresses |
| interrupt system | masked interruption; unmasked interruption |
| control in wait mode | available |
| level of external signals | meets TTL level |
| storage: main | 1K bytes, LSI, NMOS technology |
| ROM | 1K bytes, LSI, NMOS technology |
| counter and time-of-day clock (CTC): | |
| mode of operation | counter and time-of-day clock are set by program |
| number of channels | 4 |
| parallel I/O (PIO): | |
| number of channels | 2 bidirectional channels of interface (4 bits) |
| number of modes of operation | 4 |
| multiprocessor system: | |
| number of computers joined | 1 main and 3 subordinate |
| maximum data transfer rate, bytes/s | 30K |

The structure of the K-1520 microcomputer is the maximum configuration that can be reduced to one central processor without a bus, placed on a printed circuit board together with an additional maximum of 4K bytes of storage. The control console is used as an additional means for using the computer as an open module during start-up for oepration, maintenance and searching for errors and when testing programs.

The computer functional units are placed on 215 x 170 mm cards and can be assembled in the form of an open module. The microprocessor bus is made in the form of a printed circuit board on the rear panel.

Software for this microcomputer includes a macro assembler. To solve user problems, there are a number of standard modules, for example I/O modules, arithmetic and converting modules and the standard mathematical functions. There are compilers for the higher programming languages: CDL, BASIC and COBOL.

There are different operating systems for the areas of application listed above.

The SM-6307 (Robotron 1154) alphanumeric printer is a multipurpose device for output of data in the form of numbers, letters and special characters that operates with the single-digit mosaic printing principle. The device (fig. 2) is used mainly in medium-size data processing systems as a device for data collection and office mechanization, a device for output of measuring data, and in terminals and other general-purpose data I/O units.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



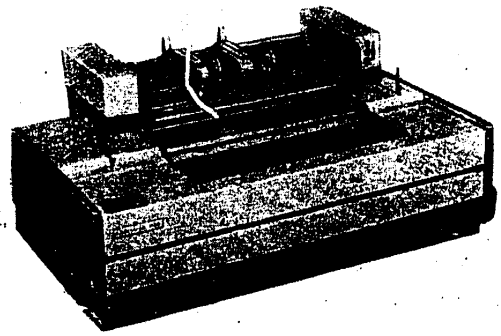


Fig. 2. SM-6307 alphanumeric printer

Specifications:

| | |
|---|---|
| printing rate, cps | 25 in start-stop mode; 45 in continuous mode |
| number of characters per line | 132 |
| character pattern | 7 x 5; expansion to 10 x 5 is possible |
| number of characters | 64 (96) |
| space between characters, mm | 2.54 |
| space between lines, mm | 4.23 |
| width of print roller, mm | 385 |
| number of print copies | 3 . . . 5 |
| paper feed technique | roll paper feed (automatic pulling of edges, manual setting of form, one linefeed for roll paper, n linefeed for perforated paper) |
| ink ribbon | width 13 mm, capable of black and red printing |
| main voltage | 220V + 10%<br>- 15% |
| main frequency | 50 Hz + 1 |
| power consumption | about 150 VA |

The SM-6307 printer contains electronic equipment, power supply, carriage with control, step motor, paper feed unit and ink ribbon transport unit.

The printer can be delivered as a spearate unit for installation in existing equipment or as a table-top model with its own housing. Dimensions of the table model with the paper feed unit are 744 x 475 x 210 mm.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



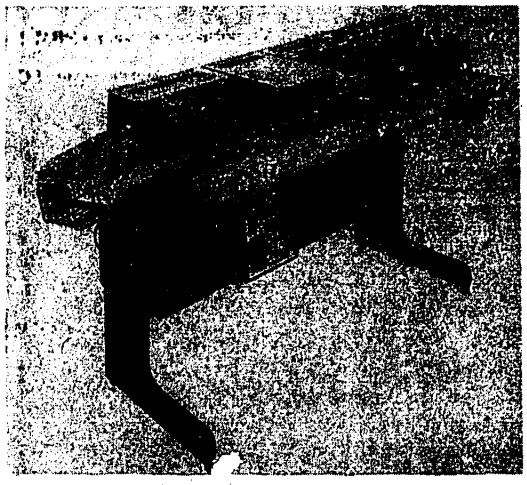


Fig. 3. SM-6903 mark reader with automatic form feed and sorter

The SM-6903 mark reader (Robotron 1375) is a standalone programmable electronic unit for data acquisition (fig. 3). The device operates on the basis of opto-electronic reading of marks. This method permits automatic recognition of source text characters. When the source forms are prepared, a marker is used manually to mark the positions that must be read and converted to particular data automatically.

This device should be applied where a large quantity of data is encountered that is distinguished by simplicity in coding. This results in savings of resources and manpower and data is rapidly prepared on media.

Specifications:

| | |
|---|---|
| transport rate | 400 mm/s |
| read rate | 2000 . . . 4000 forms/hr |
| storage and control unit | modification of the K-1510 microcomputer |
| data output | YeS-5091 cassette tape recorder |
| main voltage | 220V + 10%<br>- 15 % |
| power consumption | 340 VA |
| form size | 99 x 148 . . . 210 x 304.8 mm |
| space between rows | variable, minimum of 5 mm |
| space between columns | 8 mm |
| marker size | about 0.5 x 5 mm |
| marking medium | pencil, black ball point pen, black print |
| data media | magnetic cassette, ECMA [European Computer Manufacturers Association] 34, 0.15 inch |



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



| | |
|---|---|
| recording method | KOI [information interchange code]-TS 97/S 11/35 or ECMA 34 |
| recorded code | KOI-7 or ECMA 6 |
| a complete checking system is provided to ensure reliability and checking of data | system hardware checking (magnetic tape error, read unit, power interruption); forms check (odd marking, skew of forms, row error); program facilities check (check of field dependence, check of positions, check of check numbers, internal checks). |

The unit is supplied in two ways: hard programmed version (basic unit) and a programmable version (basic unit with expanded storage). Both types can be equipped with the automatic forms feed and sorter. The mark reader is a table model and consists of the read unit and transport, electrical portion, storage unit (YeS-5091 cassette tape recorder) and table.

Data may also be output on perforated tape using KOI-7 code. The perforator is positioned separately.

COPYRIGHT: Izdatel'stvo "Finansy i statistika", 1981

8545
CSO: 8144/0138



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



MPL/600 ALGORITHMIC LANGUAGE

Moscow VYCHISLITEL'NAYA TEKHNIKA SOTSIALISTICHESKIKH STRAN in Russian No 9, 1981 (signed to press 21 May 81) pp 161-165

[Article by A. Aleksandrov and R. Yefremova, both engineers, and Z1. Aleksandrova, candidate of engineering science, all from People's Republic of Bulgaria]

[Text] MPL/600 is a subset of the PL/1 algorithmic language. The MPL/600 compiler, developed as a cross-compiler, operates with the Unified System of Computers. The minimum computer configuration required for the compiler is: the YeS-2020 CPU, 128K bytes of main storage, the YeS-5052 or YeS-5062 disk storage unit, the YeS-7031 printer, the YeS-6012 card reader and the YeS-5012 tape drive.

The compiler consists of an absolute module cataloged in the absolute module library (SYSCLB) and a set of subroutines cataloged in the M-sublibrary of the source statement library (SYSSLB). The compiler is written in FORTRAN. Compiler speed is satisfactory: 500 MPL/600 statements are processed in about 20 minutes on a YeS-1020 computer.

MPL/600 Features and Capabilities. An MPL/600 source program is a series of statements compiled according to the language syntactic rules. During compilation, the compiler analyzes the source program statements and translates them into the Assembler language for the SM-600, called an object program. This program can be translated by using the Assembler in the SM-600 family of microprocessors and executed on a microprocessor system. The MPL/600 compiler detects syntactic errors in the source program and puts out the appropriate messages, after which an equivalent group of SM-600 Assembler instructions is generated for each statement.

Source program statements must be punched on cards in columns 1 to 72. If the statement does not fit on one card, it can be continued on successive cards, except real constants. Multiple blanks are ignored by the compiler. Blanks in literal data are preserved. Logic statements and relational operations must be enclosed by two blanks. Each MPL/600 statement may have a label. A label has no more than six characters, and the first must be a letter. Comments may be included at will in the program for explanations.

The compiler provides the capability of including in the source program groups of statements written in Assembler for the SM-600. Each statement of this type must have a $ in column 1. Each MPL/600 statement performs one of the following



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



functions: specifies characteristics of source program and type of data with which the compiler operates, and causes execution of specified operations. MPL/600 statements are usually compiled from key words that are used together with basic elements in the language: constants, variables and expressions. Key words may not be used as labels. There are several basic groups of MPL/600 statements.

Arithmetic statements provide the capability of executing arithmetic operations and replacing the current value of a referenced variable by the result. The following arithmetic operations are permitted in the language:

| | |
|---|---|
| SHIFT | shift (arithmetic left or right) |
| IAND | conjunction |
| IOR | disjunction |
| IEOR | modulo 2 addition |
| X | multiplication |
| / | division |
| + | addition |
| - | subtraction. |

Arithmetic statements are executable statements.

Control statements give the user the capability of controlling the execution of the object program. The folowing control statements are permitted:

| | |
|---|---|
| GO TO | unconditional branch statement |
| IF | conditional branch statement |
| DO | loop statement |
| CALL | subroutine exit statement |

Control statements are executable.

Declarative statements are used to describe the type and of and storage amount needed for variables and arrays. The following data types are permitted:

| | |
|---|---|
| BIT | bit string |
| BINARY | binary integers |
| DECIMAL | decimal numbers |
| SIGNED DECIMAL | decimal number with sign |
| CHARACTER | character string. |

All variables must be explicitly declared in the source program.

Procedural statements allow a user to define logically independent program units that can be compiled separately or together with the main program.

The MPL/600 language provides the capability of working with three-dimensional arrays (no more than three subscripts).

Special attention is paid to pointers, the use of which facilitates processing of unidimensional arrays to a considerable extent.

The MPL/600 library consists of 32 built-in subroutines, abou 100 subroutines for operating with arrays and over 100 user subroutines.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



The built-in subroutines are used by the compiler in translating basic arithmetic operations (addition, subtraction, multiplication, division, decimal arithmetic, data conversion, etc.). The user can also call these subroutines.

Array operation subroutines are used for basic operations with arrays: assignment; addition, subtraction, multiplication and division of all array elements by a constant; member-by-member addition, subtraction, multiplication and division of two arrays with an identical dimension, etc.

Subroutines ensure high speed with the minimal size of required main storage. Library subroutines are called by using the CALL statement and parameters. For example, CALL ZF50 (P, Q, N), where ZF50 is the subroutine name, P is the array of the result, Q is the source array and N is the dimension of the arrays.

Library subroutines can be placed in permanent storage. With RAM type main storage they are not needed. All subroutines are placed in the source statement library (SYSSLB) in the Unified System computer, from where they are called as needed by the compiler, which places them at the end of an object program. They are compiled in Assembler language for the SM-600.

YeS DOS facilities give the user the capability of cataloging his MPL/600 or SM-600 Assembler programs in the library. Each of them can consist of a practically unlimited number of instructions. The maximum number of subroutines in a library is 236.

MPL/600 and PL/1 Differences. The differences between these two algorithmic languages boils down to the difference in notation on the input medium, description of data, statements and compilation facilities.

Differences in notation on the input medium:
constants must be written on one line;
a label may have no more than six characters;
the presence of comments is defined by the character ] in front of them;
a $ in column 1 indicates a statement in Assembler for the SM-600.

Differences in data description:
the FLOAT attribute is not provided for, i.e. in DECIMAL and BINARY declarations, FIXED is implied;
the declaration DECIMAL(m, n) corresponds to PICTURE'(m-n-1)9.(n)9,
lengths of variables must be within the following bounds:

| Type of Variable | m | n | Unit of Measurement |
|---|---|---|---|
| BIT | 1-8 | | bit |
| BINARY | 1-2 | | byte |
| DECIMAL | 1-12 | 1-12 | digit |
| CHARACTER | 1-256 | | character |

where m is the total number of characters and n is the number of digits after the point.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



Differences in statements and operations:
logic operations and compare operations must be notated as follows:

| | |
|---|---|
| NOT | negation |
| AND | logical AND |
| OR | logical OR |
| EQ | = |
| NE | ≠ |

GT—>;
GE—≥;
LT—<;
LE—≤;

the SHIFT operation does not exist in PL/1;
IAND, IOR and IEOR operations are considered arithmetic operations and are executed bit by bit;
MPL/600 has a computable GO TO statement;
in MPL/600, there is no DO statement with a list of values.

Differences in compilation facilities are:
in the PROCEDURE statement, one can write the name of a stack after the attribute MAIN;
the main procedure does not have to have a name.

As can be seen, these differences are insignificant, since programs compiled in MPL/600 can also be translated by a PL/1 compiler.

Example. The figure shows an MPL/600 program for computing the sum of elements of an array whose values are no more than 50. The text of the object program contains both MPL/600 statements written as comments in Assembler for the SM-600 and a series of instructions in Assembler for the SM-600 generated for each statement.

Using the MPL/600 algorithmic language makes it considerably easier to compile complex programs for microprocessor systems oriented to the SM-600. The existing convenient link between MPL/600 and Assembler for the SM-600 provides for assembly of an object program and obtaining perforated tape in machine language. The set of library subroutines can be expanded by new subroutines that are called by the CALL statement. The library is expanded by standard YeS DOS facilities.



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6



```
00001 J        OPT   TAPE
00002  PROCEDURE OPTIONS(MAIN);
       RMB  40
000    LDS  #T+39
00003        DECLARE S BINARY(2),
       JMP  Z001
       RMB  2
00004                I BINARY(1),
       RMB  1
00005          MAS(100) BINARY(1);
AS     RMB  100
00006         S=0;
001    LDX  Z425
       STX  S
00007         DO I=1 TO 100;
       LDAA #1
002    STAA I
00008            IF MAS(I) GT 50 THEN GO TO NEXT;
       LDX  #MAS
       LDAB I
       JSR  ZF1F
       LDAA 0,X
       CMPA #50
       BLE  Z004
       JMP  NEXT
00009              S=S+MAS(I);
004    LDX  #MAS
       LDAB I
       JSR  ZF1F
       LDAA 0,X
       CLRB
       TSTA
       BGE  *+3
       COMB
       ADDA S+1
       ADCB S
       STAB S
       STAA S+1
00010  NEXT: END;
EXT    LDAA I
       CMPA #100
       BCC  Z005
       INCA
       JMP  Z002
00011         END;   /* КРАЙ НА ПРОГРАМАТА */
005    EQU  *
425    FDB  0
       PAGE
F1F      BNE   <F1F2  *** ADD TO INDEX ***                    903
         DEX                                                  904
         RTS                                                  905
F1F2     DECB                                                 906
F5F      STX   I+6                                            907
         ADDB  I+7                                            908
         STAB  I+7                                            909
         BCC   <F1F1                                          911
         INC   I+6                                            912
F1F1     LDX   I+6                                            913
         RTS                                                  914
FFF    EQU   *
       END
       EOM
```

Program in the MPL/600 language for computing the summ of elements

COPYRIGHT: Izdatel'stvo "Finansy i statistika", 1981

8545
CSO: 8144/0138

END



APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040033-6